

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Web-орієнтована інформаційна система страхового агентства
Назва теми

КВРІСТ 190119.21.01.03 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконав: студент IV курсу, група ICT-20-1


Підпис

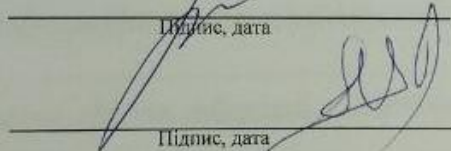
В.В. Катренюк
Ініціали, прізвище

Керівник


Підпис, дата

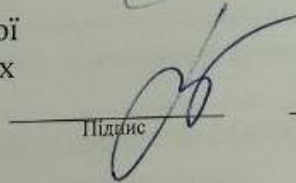
Є.Г. Гнатчук
Ініціали, прізвище

Нормоконтролер


Підпис, дата

С.М. Лисенко
Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем


Підпис

Т.О. Говорущенко
Ініціали, прізвище

«24» червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорущенко

“ 10 ” 01 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Катренюку Владиславу Валерійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Web-орієнтована інформаційна система страхового агентства

Керівник проекту (роботи) Гнатчук Є.Г., к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2024 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Web-орієнтована інформаційна система страхового агентства та постановка задачі щодо її удосконалення

Проектування системи обробки інформації у Web-орієнтованій інформаційній системі страхового агентства

Реалізація клієнтської та серверної частини у Web-орієнтованій інформаційній системі страхового агентства

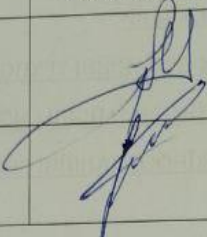
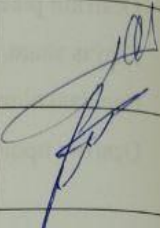
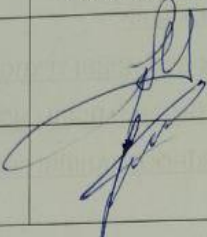
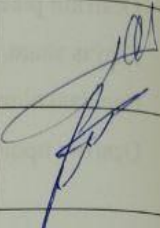
5. Перелік структур систем та алгоритмів функціонування (із зазначенням обов'язкових креслень) _____

Структури та UML-діаграми інформаційної системи

Підсумковий аналіз тестувань та рекомендацій

Скріншоти інформаційної системи

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., д.т.н., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2024 р.

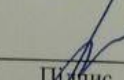
КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – вибір компонентів для проектування інформаційної системи страхового агентства	01.04.2024	виконано
5	Робота над розділом 3 – проектування інформаційної системи страхового агентства та її тестування для злагодженої роботи	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	26.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент


Підпис

Керівник роботи


Підпис

В.В. Катренко
Ініціали, прізвище

С.Г. Гнатчук
Ініціали, прізвище

№
р
я
д
к
а





1

2

3

4

№ рядка	Формат	Позначення	Найменування	Кількість	№ екз	Примітка
1		КВРІСТ 190119.21.01.03 ПЗ	Текстові документи Пояснювальна записка	58		
2		КВРІСТ 190119.21.01.03 Е8	Графічні матеріали Структури та UML-діаграми інформаційної системи	1		
3		КВРІСТ 190119.21.01.03 Е8	Підсумковий аналіз тестувань та рекомендацій	1		
4		КВРІСТ 190119.21.01.03 Е8	Скріншоти інформаційної системи	1		

КВРІСТ 190119.21.01.03 ВП				
Зм	Арк	№ докум	Підпис	Дата
Розробив		Катренко		
Перевір.		Гнатчук		
Н. контр.		Лисенко		
Затв.		Говорушення		04.06

Розвиток інформаційних систем у сфері страхового агентства. Відомість проекту.	Літера	Аркуш	Аркушів
	У	1	1
ХНУ, ІСТ-20-1			

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Web-орієнтована інформаційна система страхового агентства».

Автор роботи: Катренюк Владислав Валерійович.

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 58 с., 16 мал., 2 табл., 3 дод., 60 джерел.

Клієнтська частина, Серверна частина, Контент, База Даних.


Метою дипломної роботи є підвищення ефективності роботи страхового агентства шляхом створення інформаційної системи.

Об'єктом дослідження є принципи побудови інформаційної системи страхового агентства.

Предметом дослідження є інформаційна система страхового агентства.

Практичне значення має спроектована та реалізована система, яка покращує процеси управління ризиками, ефективність страхових стратегій та якість обслуговування клієнтів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.



Підпис студента





30.05.2024

Дата

ЗМІСТ

ВСТУП		4
1 WEB-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА СТРАХОВОГО АГЕНТСТВА ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ НОРМАЛЬНОГО ФУНКЦІОНУВАННЯ		5
1.1 Аналіз інформаційної системи страхового агентства		5
1.2 Вибір інформаційного забезпечення Web-орієнтованої інформаційної системи страхового агентства		6
1.2.1 Перегляд існуючих інформаційних систем страхової компанії		6
1.2.2 Оцінка сучасних засобів створення інформаційних систем у напрямку страхового агентства		7
1.3 Постановка задачі Web-орієнтованої інформаційної системи страхового агентства		10
1.3.1 Аналіз вимог		10
1.3.2 Проектування		11
1.3.3 Розробка функціоналу		12
1.3.4 Вебдизайн		13
1.3.5 Технічна реалізація		14
1.3.6 Інфологічне проектування бази даних		15
1.3.7 Тестування		18
1.3.8 Оптимізація для пошукових систем		18
1.3.9 Розробка функціоналу		18
1.4 Висновки		23
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ		24
2.1 Архітектура інформаційної системи		24
2.1 Функціональні вимоги до інформаційної системи		33
2.3 Нефункціональні вимоги до інформаційної системи		33
2.4 Користувацькі вимоги до інформаційної системи страхового агентства		35
2.5 Висновки		36

КвРІСТ 190119.21.01.03 ПЗ

Зм.	Арк.	Недокум.	Підпис	Дата		Літера	Аркуш	Аркущів
Виконав		Катренко В.В.			Розвиток інформаційних систем у сфері страхового агентства. Пояснювальна записка.	у	2	58
Перевір.		Гнатюк С.Г.				ХНУ ІСТ-20-1		
Н.контр.		Лисенко С.М.						
Затвер.		Говорушенко Т.О.		24.05				

3.2.2 Використання архітектурного патерну Модель-Вид-Шаблон (MVT) у Node.....	43
3.3 Огляд ключових модулів системи.....	45
3.4 Тестування ефективності системи.....	52
3.5 Рекомендації щодо подальшого розвитку системи	55
3.6 Висновки	57
ВИСНОВКИ	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	60
ДОДАТОК А КОПІЯ КРЕСЛЕННЯ «СТРУКТУРИ ТА UML-ДІАГРАМИ ІНФОРМАЦІЙНОЇ СИСТЕМИ».....	66
ДОДАТОК Б КОПІЯ КРЕСЛЕННЯ «ПІДСУМКОВИЙ АНАЛІЗ ТЕСТУВАНЬ ТА РЕКОМЕНДАЦІЙ».....	67
ДОДАТОК В КОПІЯ КРЕСЛЕННЯ «СКРІНШОТИ ІНФОРМАЦІЙНОЇ СИСТЕМИ».....	68

ВСТУП

Актуальність дослідження. Інформаційна система страхового агентства відіграє ключову роль у ефективному функціонуванні та розвитку компанії. Основна функція страхової компанії полягає у наданні захисту від різних ризиків та втрат, пов'язаних із життям, здоров'ям, майном, транспортом та іншими аспектами життя та діяльності.

Інформаційна система страхового агентства є ресурсним порталом, призначеним для ознайомлення клієнтів з послугами, які пропонує компанія, а також для проведення операцій з оформлення страхових полісів.

Метою дипломної роботи є підвищення ефективності роботи страхового агентства шляхом створення інформаційної системи.

Об'єктом дослідження є принципи побудови інформаційної системи страхового агентства.

Предметом дослідження є інформаційна система страхового агентства.

Практичне значення має спроектована та реалізована система, яка покращує процеси управління ризиками, ефективність страхових стратегій та якість обслуговування клієнтів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						4
Арк.	№ докум.	Підпис	Дата			

1 WEB-ОРІЄНТОВАНА ІНФОРМАЦІЙНА СИСТЕМА СТРАХОВОГО АГЕНТСТВА ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО ЇЇ НОРМАЛЬНОГО ФУНКЦІОНУВАННЯ

1.1 Аналіз інформаційної системи страхового агентства.

Інформаційна система страхового агентства є організаційною структурою, яка займається наданням страхових послуг клієнтам. Система налічує у собі різні види страхування, такі як автомобільне, життєве, нерухомість, туристичне та інші, які пропонуються клієнтам. Агентство зазвичай працює з кількома страховиками, щоб мати можливість запропонувати різноманітні продукти.

Страхування є складним та важливим елементом сучасного життя, що відіграє ключову роль у фінансовому плануванні та захисті від ризиків. Страхування надає захист від фінансових втрат, пов'язаних з непередбачуваними подіями. Компанія працює за складною та виваженою схемою, яка включає в себе різні етапи та процеси. Для початку відбувається реєстрація та укладення договору, для початку потрібно обрати страхування яке найкраще відповідає вашим потребам. Це може включати вибір типу страхування і визначення необхідних умов покриття. Заповнюється заявка на страхування, в якій потрібно надати інформацію про себе, про об'єкт страхування, а також про попередні страхові випадки. Страхова компанія оцінює ризик та ймовірність страхових випадків на основі наданої інформації. Якщо пропозиція відповідає вашим очікуванням, ви приймаєте умови та підписуєте договір на страхування. Після прийняття умов, ви зазвичай зобов'язані сплатити премію за страховку.

Страховики використовують статистичні дані, математичні моделі та свій досвід для оцінки ймовірності виникнення різних ризиків. На основі оцінки ризиків страхова компанія визначає вартість страховки або премію. На основі всієї зібраної інформації та оцінки ризиків страхова компанія приймає рішення про надання страхового покриття, його обсяг та умови.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						5
Арк.	№ докум.	Підпис	Дата			

У разі настання страхового випадку клієнт повідомляє страхову компанію та подає всю необхідну документацію. Страхова компанія реєструє заявку на обробку страхового випадку у своїй системі, надає їй унікальний номер і починає процес обробки. Аналізує надані документи та збирає додаткову інформацію, якщо необхідно, для перевірки справжності страхового випадку та оцінки розміру збитків. На основі наданої інформації страховики оцінюють розмір збитків та визначають, чи підпадають вони під умови страхового полісу.

Якщо заявку на відшкодування збитків схвалено, страхова компанія виплачує страхове відшкодування страхувальнику відповідно до умов договору. Після того, як компенсацію виплачено, страхова компанія складає звіт про виплату. У разі незгоди з рішенням страхової компанії страхувальник має право на оскарження.

Існує ще такий продукт компанії як перестраховування, при якому страхова компанія передає частину ризику, який вона приймає від своїх клієнтів, іншій страховій компанії або групі страхових компаній, які називають перестраховиками. Це робиться з метою захисту від великих збитків та забезпечення фінансової стабільності. Після того, як страхова компанія визначила частину ризику для передачі, вона обирає перестраховиків, з якими співпрацюватиме. При необхідності страхова компанія може оновлювати або переглядати свої договори з перестраховиками, щоб адаптувати їх до умов ринку, що змінюються, або власним потребам.

1.2 Вибір інформаційного забезпечення Web-орієнтованої інформаційної системи страхового агентства

1.2.1 Перегляд існуючих інформаційних систем страхової компанії

У данній роботі буде приведений аналіз такої інформаційної системи страхової компанії як "Ingo.ua". Інтернет-сайт "Ingo.ua" є важливим ресурсом для українських користувачів та має ряд функцій, можливостей та деталей,

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						6
Арк.	№ докум.	Підпис	Дата			

спрямованих на надання різноманітних послуг та інформації. "Ingo.ua" пропонує різноманітні види страхування для фізичних та юридичних осіб. Це включає автострахування, медичне страхування, страхування майна, подорожей, відпочинку та багато іншого. Сайт надає доступ до різноманітних онлайн-калькуляторів для розрахунку вартості страхових полісів. Це дозволяє користувачам швидко та зручно отримати інформацію про страхові тарифи та умови. На сайті можливе створення особистого кабінету для користувачів, де вони можуть здійснювати управління своїми полісами, переглядати історію оплат та отримувати доступ до інших персоналізованих послуг. Сайт може бути доступним через мобільні пристрої, що дозволяє користувачам отримувати доступ до сервісів страхування з будь-якого місця та в будь-який час. "Ingo.ua" надає широкий спектр страхових послуг та інформаційних ресурсів для користувачів, спрямованих на забезпечення їхнього захисту та комфорту.

1.2.2 Оцінка сучасних засобів створення інформаційних систем у напрямку страхового агентства

У розробці даної інформаційної системи для страхового агентства вирішено використовувати такі сучасні технології як JavaScript та React для фронтенду, а також HTML та CSS для створення та стилізації користувацького інтерфейсу. JavaScript є універсальною мовою програмування, яка дозволяє реалізувати складні функції на вебсторінках, а фреймворк React відомий своєю ефективністю та зручністю у створенні інтерактивних користувацьких інтерфейсів. Це комбінація технологій надає нам можливість створити швидко та відгукову вебзаплікацію, яка може ефективно масштабуватися та адаптуватися до змінних вимог бізнесу.

JavaScript є ключовим інструментом у світі розробки вебсайтів і вебзастосунків, зокрема для створення інформаційних систем страхових агентств. Як мова програмування, що виконується на стороні клієнта, JavaScript дозволяє

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 7
Арк.	№ докум.	Підпис	Дата			

реалізувати інтерактивність і динаміку вебсторінок, обробляти події користувацького інтерфейсу, виконувати асинхронні запити до сервера та маніпулювати DOM для оновлення контенту без перезавантаження сторінки.

В сучасній розробці JavaScript широко використовується не тільки для клієнтської частини, але й для серверної логіки завдяки Node.js — це оточення виконання коду JavaScript на сервері, що включає підтримку великої кількості модулів та пакетів через npm. Це робить JavaScript універсальним рішенням для повного стеку веброзробки, дозволяючи розробникам використовувати одну мову як для фронтенду, так і для бекенду.

React — це декларативна JavaScript-бібліотека для створення користувацьких інтерфейсів, що надає ефективні шляхи для розробки компонентно-орієнтованих вебзастосунків. Використання React сприяє швидкому реагуванню додатка на дії користувача та забезпечує гладку взаємодію без зайвих перезавантажень сторінки, що підвищує загальну продуктивність і зручність використання вебсайтів.

Завдяки можливостям сучасних Single Page Applications (SPA) та JavaScript-фреймворків, таких як React, Angular чи Vue.js, розробка інформаційних систем страхових агентств набуває нових горизонтів. Реалізація комплексних користувацьких сценаріїв, інтеграція з різноманітними API та побудова масштабованих і безпечних вебзастосунків з високою швидкістю відгуку стає не тільки можливою, але й більш доступною.

HTML (HyperText Markup Language) - це стандартна мова розмітки для створення вебсторінок. Вона використовується для структурування змісту вебсторінок та визначення різноманітних елементів, таких як текст, зображення, відео, посилання і форми. HTML дозволяє вбудовувати інші мови програмування, такі як JavaScript та CSS, для створення динамічних та стильових ефектів. HTML складається з набору тегів, які вказують браузеру, як відобразити вміст сторінки.

HTML є основною мовою для створення вебсторінок та є основою веброзробки разом з CSS і JavaScript. Вона забезпечує структуру та зміст

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 8
Арк.	№ докум.	Підпис	Дата			

вебсторінок, що дозволяє створювати різноманітний та інформативний вебконтент.

CSS (Cascading Style Sheets) - це мова опису стилів, яка використовується для визначення вигляду та оформлення вебсторінок, написаних з використанням HTML та інших мов розмітки. CSS визначає, які кольори, шрифти, розміри, відступи, вирівнювання та інші стилістичні властивості будуть застосовані до елементів сторінки. CSS використовує селектори для вибору елементів HTML, до яких будуть застосовані стилі.

CSS дозволяє створювати естетично приємний та привабливий вебдизайн, а також забезпечує можливість розмітки та форматування вмісту сторінок для зручності користувачів. Він є невід'ємною частиною веброзробки разом з HTML та JavaScript.

Сервер і домен це дві ключові складові, які використовуються для розміщення та доступу до вебсайтів.

Сервер - це комп'ютер або програмне забезпечення, яке надає послуги зберігання, він відповідає за обробку запитів користувачів і відправку відповідей у вигляді вебсторінок та іншого контенту. Коли ви відкриваєте вебсторінку у браузері, запит на цю сторінку надсилається на сервер, який потім обробляє цей запит і надсилає вебсторінку назад у ваш браузер для відображення. Сервер зберігає всі файли вебсайту, такі як HTML-сторінки, зображення, відео, стилі CSS, скрипти JavaScript та інші ресурси.

Домен сайту - це унікальна адреса, яка використовується для ідентифікації конкретного вебсайту в Інтернеті. Домен дозволяє визначати унікальне місцезнаходження вебсайту в мережі Інтернет і забезпечує зручний та легкодоступний спосіб доступу до нього. Домен використовується в URL-адресах, що дозволяє користувачам легко та швидко знаходити вебсайт.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						9
Арк.	№ докум.	Підпис	Дата			

1.3 Постановка задачі Web-орієнтованої інформаційної системи страхового агентства

1.3.1 Аналіз вимог

Для аналізу вимог інформаційної системи страхової компанії необхідно враховувати різноманітні аспекти, пов'язані з бізнес-процесами та операціями страхування.

Головні вимоги до системи:

- реєстрація та редагування нових полісів, страхових випадків;
- облік статусу полісів;
- оцінка збитків та розрахунок страхових виплат;

Клієнтська база даних:

- облік клієнтів;
- зберігання та оновлення даних про страхові поліси;
- управління історією страхових випадків;

Вимоги до безпеки:

- захист конфіденційності даних клієнтів та шифрування даних;
- резервне копіювання та відновлення даних;

Інтеграція:

- інтеграція із системами страхових брокерів;
- інтеграція з бухгалтерськими програмами;
- обмін даними із платіжними системами;

Додаткові процедури для продуктивності:

- використання індексів та інших методів для прискорення виконання запитів;
- використання кешування даних для зберігання даних;
- проведення регулярного тестування продуктивності системи;

Рекомендації:

- створення інтуїтивної навігації та структури меню;

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 10
	Арк.	№ докум.	Підпис	Дата		

- надання декілька мов інтерфейсу;
- забезпечення цілодобової технічної підтримки;

Вимоги до аналітики та звітності:

- стандартні звіти про продаж, виручку, збитки;
- можливість створювати аналітичні звіти;
- автоматичне оновлення даних;

1.3.2 Проектування

Проектування інформаційної системи для страхового агентства включає кілька ключових етапів, спрямованих на створення ефективної і надійної системи, здатної керувати всіма аспектами бізнесу страхування.

Для початку потрібно визначити процеси які включають вивчення поточних процесів страхового агентства, виявлення його потреб і проблем, а також визначення цілей, які повинні бути досягнуті за допомогою нової інформаційної системи.

На основі аналізу процесів визначаються функціональні та нефункціональні вимоги до інформаційної системи. Це включає визначення функціональності, інтерфейсів, безпеки, продуктивності та інших аспектів системи. Вивчення існуючих документів, процедур та робочих процесів у страховому агентстві допоможе зрозуміти поточний стан справ, виявити вузькі місця та області для покращення системи.

Проектування бази даних для страхового агентства має бути ретельно продуманим і ґрунтується на специфікації вимог до системи. Необхідно створити структуру бази даних, яка зберігатиме всю необхідну інформацію про клієнтів, поліси, страхові випадки тощо. На основі аналізу потрібно визначити всі сутності та їх атрибути. Переконайтесь, що всі сутності та їх атрибути явно визначені.

На основі вимог розробляється програмне забезпечення, яке оброблятиме дані, автоматизуватиме процеси, надаватиме інтерфейси для користувачів тощо.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 11
	Арк.	№ докум.	Підпис	Дата		

Це включає створення діаграм класів, діаграм бази даних, проектування інтерфейсів користувача та інші документи проектування.

Після розробки проводиться тестування системи для перевірки її працездатності, безпеки та відповідності до вимог. Це включає функціональне тестування, тестування продуктивності, тестування безпеки. Після успішного завершення тестування система впроваджується у робоче середовище страхового агентства.

1.3.3 Розробка функціоналу

Розробка функціоналу інформаційної системи для страхового агентства має бути спрямована на задоволення основних потреб бізнесу та забезпечення ефективного управління усіма аспектами страхової діяльності. Функціональні компоненти можуть бути доповнені та адаптовані залежно від специфіки бізнесу та вимог страхової агенції. Важливо також забезпечити інтеграцію інформаційної системи із зовнішніми системами.

Перш за все потрібно мати метод додавання та видалення клієнтів із бази даних, а також зберігання на оновлення їх інформації у цій же базі даних, розробити можливість зберігання різноманітних операцій будь-якого характеру доступною для кожної сторони системи, як для клієнтів так для агентів.

Головний атрибут системи це створення та управління полісами, у системі має бути вбудований калькулятор підрахунку премій та страхових внесків. Ці поліси обов'язково мають бути захоплені методом, який буде показувати їх статус. Система має мати змогу реєструвати та відстежувати страхові випадки, можливість оцінювати збитки та обробляти виплати агентами.

Реєструвати та відстежувати страхові випадки, спроектувати оцінку збитків та обробки виплат, це важливий компонент системи це взаємодія з клієнтами та іншими зацікавленими сторонами у процесі врегулювання збитків.

Фінанси:

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 12
Арк.	№ докум.	Підпис	Дата			

- облік фінансових операцій;
- фінансова звітність та аналіз;
- інтеграція із системами платіжних шлюзів для обробки платежів;

Управління агентами та їх аналітика:

- реєстрація та управління агентами та їх унікальними профілями;
- підтримка спільної роботи та обміну інформацією між агентами;
- генерація статистичних звітів про продажі, страхові випадки;

Ще що я би хотів додати до системи так це управління доступом до даних та функціоналом системи залежно від ролі користувача, та забезпечення захисту даних та конфіденційності клієнтів за допомогою шифрування, автентифікації та інших заходів безпеки.

1.3.4 Вебдизайн

Дизайн інформаційної системи страхового агентства має бути інтуїтивно зрозумілим, зручним для використання та естетично привабливим. Меню, навігація та елементи керування мають бути легко доступними та зрозумілими.

Потрібно структурувати інформацію таким чином, щоб користувачі могли легко знайти те, що їм потрібно. Використовуйте категорії, фільтри та сортування для спрощення пошуку та навігації. Інтерфейс має бути ясным та зрозумілим для користувачів будь-якого рівня навичок. Використовувати тільки зрозумілі та консистентні метафори, іконки та текстові підказки.

Необхідно створити приємний вигляд для системи, використовуючи та враховуючи корпоративний стиль і бренд страхової агенції. Одне із головних це гнучкість інтерфейсу, щоб система коректно відображалася на різних пристроях та екранах, включаючи комп'ютери, планшети та смартфони. Оптимізувати форми для введення даних, щоб користувачі могли їх легко заповнювати. Використовуйте валідацію даних та підказки для допомоги користувачам під час заповнення форм.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 13
Арк.	№ докум.	Підпис	Дата			

Надати можливість персоналізації інтерфейсу залежно від переваг користувачів. Наприклад, користувачі можуть налаштовувати видимість та порядок відображення інформації. Забезпечити доступність інтерфейсу для користувачів з обмеженими можливостями, включаючи можливість збільшення розміру шрифтів, використання скринридерів, це дуже важливо.

Особливу увагу приділити захисту даних та безпеки користувачів під час розробки вебінтерфейсу. Бажано використовувати SSL-шифрування, механізми автентифікації та авторизації та інші заходи безпеки.

1.3.5 Технічна реалізація

Для розробки інформаційної системи страхового агентства я обрав комплекс технологій, які оптимізують процес створення та подальшого функціонування проєкту. Використання вебтехнологій HTML та CSS забезпечує створення структури та оформлення користувацького інтерфейсу, тоді як JavaScript разом з бібліотекою React дають можливість розробити інтерактивну клієнтську частину. Бекенд-частина реалізована за допомогою Node.js, який дозволяє використовувати JavaScript і для серверної логіки, забезпечуючи таким чином єдиність мови програмування по всьому стеку технологій.

Архітектура системи базується на клієнт-серверній моделі, де Node.js використовується для обробки логіки додатку, запитів користувачів та управління даними. HTML та CSS, у свою чергу, забезпечують створення зручного і сучасного користувацького інтерфейсу, а React допомагає у розробці компонентно-орієнтованих вебзастосунків, які з легкістю можуть масштабуватися та адаптуватися до потреб користувачів.

У якості системи управління базами даних може використовуватися MongoDB або інша NoSQL база даних, яка добре інтегрується з JavaScript-орієнтованим стеком технологій та підтримує гнучкість у роботі з даними. Це

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 14
Арк.	№ докум.	Підпис	Дата			

дозволяє швидко розгортати структури баз даних, які можуть ефективно масштабуватися та легко модифікуватися з ростом проєкту.

Систематичне тестування на кожному етапі розробки, від модульних тестів до комплексного системного тестування, забезпечує надійність та продуктивність системи. Розгортання проєкту передбачає використання сучасних сервісів та платформ, які сприяють легкому масштабуванню та високій доступності системи. Вебсервери на базі Node.js забезпечують швидке та ефективне виконання програмного коду, а інтеграція з React та іншими вебфреймворками відкриває широкі можливості для розширення функціоналу вебінтерфейсу.

1.3.6 Інфологічне проектування бази даних

Концептуальна модель - це сукупність взаємопов'язаних понять, що лежать в основі дослідницької системи. Це може бути простий список понять та їх можливих зв'язків. Концептуальна модель бази даних визначає основні сутності та їх взаємозв'язки у предметній галузі, не заглиблюючись у технічні деталі реалізації. На представленому рисунку (Рис. 1.1) зображено концептуальну модель інформаційної системи страхового агенства.



Рисунок 1.1 – Концептуальна модель інформаційної системи страхового агенства

Клієнти:

- ідентифікатор клієнта;
- дані клієнта;
- адреса;

Поліси:

- ідентифікатор полісу;
- тип страхування;
- страхова сума;

Страхові випадки:

- ідентифікатор випадку;
- дата випадку;
- опис події;

Платежі:

- ідентифікатор платежу;
- дані платежу;

Агенти:

- ідентифікатор агента;
- ініціали;

Етап створення логічної моделі називається логічним проектуванням. При переході від концептуальної моделі до логічної слід мати на увазі, що концептуальна модель повинна включати всю інформацію про предметну область, потрібну для проектування бази даних. Перетворення концептуальної моделі на логічну, що здійснюється за формальними правилами. Логічна модель бази даних будується в термінах інформаційних одиниць, допустимих у тій конкретній СУБД, серед якої ми проектуємо базу даних. На представленому рисунку (Рис. 1.2) зображено логічну модель даних компанії. Спроекувати логічну структуру бази даних означає визначити всі інформаційні одиниці та зв'язки між ними, задати їх імена. Якщо для інформаційних одиниць можливе

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 16
	Арк.	№ докум.	Підпис	Дата		

використання різних типів, то визначити їх тип. Слід також задати деякі кількісні характеристики, наприклад, довжину поля.



Рисунок 1.2 – Логічна модель даних компанії

Нормалізація відносин у базі даних - це процес організації структури даних відповідно до певних нормальних форм (наприклад, першої, другої та третьої нормальних форм), щоб уникнути надмірності даних та забезпечити цілісність та ефективність зберігання даних.

Кожне правило називається нормальною формою. Якщо виконується перше правило, говорять, що база даних представлена в першій нормальній формі. Якщо виконуються три перші правила, вважається, що база даних представлена у третій нормальній формі.

1.3.7 Тестування

Перевірити функціональність системи з урахуванням вимог та очікувань користувачів. Включає тестування основних процесів, таких як оформлення полісів, обробка заявок, розрахунок премій, врегулювання збитків та інші. Перевірити взаємодію між різними компонентами інформаційної системи страхової агенції. Це включає тестування інтеграції із зовнішніми системами (наприклад, платіжними системами, системами розрахунків) та внутрішніми модулями системи. Уся система має працювати як одне ціле. Перевіряється робота системи в реальних умовах використання, включаючи її продуктивність, надійність та масштабованість.

1.3.8 Оптимізація для пошукових систем

Провести дослідження ключових слів, які потенційні клієнти можуть використовувати під час пошуку інформації про страхові послуги. Використовуйте такі інструменти, як Google Keyword Planner, щоб знайти найбільш релевантні та популярні запити. Упевнитись, що кожна сторінка вебсайту має унікальні та інформативні мета-теги.

1.3.9 Розробка функціоналу

В даному підрозділі розглядається процес постановки задачі до вебзастосунку для управління фінансами. Цей додаток створений допомогати користувачам ефективно контролювати доходи та витрати, планувати та аналізувати фінансові дані. У зв'язку з цим, дослідження проблеми та визначення мети розробки є першочерговими завданнями, які забезпечать успішність проекту.

Для досягнення поставленої мети розробки вебзастосунку необхідно виконати наступні основні завдання:

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 18
Арк.	№ докум.	Підпис	Дата			

1. Розробка інтуїтивно зрозумілого інтерфейсу користувача
 - розробка адаптивного дизайну, що забезпечує зручність використання на різних пристроях;
 - створення навігаційних елементів, що дозволяють користувачам легко знаходити необхідні функції та інформацію;
 - забезпечення відповідності інтерфейсу сучасним стандартам вебдизайну та вимогам користувачів;
2. Реалізація функціоналу для ведення обліку фінансових операцій
 - створення функцій для додавання, редагування та видалення фінансових транзакцій;
 - розробка системи категоризації доходів та витрат для забезпечення зручності обліку та аналізу фінансових даних;
 - інтеграція з банківськими системами для автоматичного завантаження та оновлення транзакцій;
3. Розробка системи звітів та аналітики
 - створення функціоналу для генерації різноманітних звітів про фінансовий стан користувача, включаючи щомісячні, квартальні та річні звіти;
 - розробка аналітичних інструментів для відстеження тенденцій та прогнозування фінансових результатів;
 - забезпечення можливості експорту звітів у різні формати для подальшого використання та аналізу;
4. Забезпечення високої продуктивності та безпеки системи
 - оптимізація коду та структури бази даних для забезпечення швидкого доступу до даних та високої продуктивності вебзастосунку;
 - впровадження сучасних методів шифрування для захисту персональних даних користувачів від несанкціонованого доступу;
 - регулярне тестування системи на наявність вразливостей та усунення потенційних загроз безпеці;

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 19
	Арк.	№ докум.	Підпис	Дата		

Кожне з перерахованих завдань спрямоване на створення ефективного та зручного інструменту для управління особистими фінансами, який задовольняє потреби сучасних користувачів та відповідає високим стандартам якості та безпеки.

Функціональні вимоги:

1. Управління фінансовими транзакціями

- система повинна дозволяти користувачам додавати нові фінансові транзакції з деталями, такими як дата, сума, категорія (доходи або витрати), та опис;

- користувачі повинні мати можливість редагувати раніше додані транзакції, змінюючи їхні деталі;

- система повинна надавати можливість видалення транзакцій, які більше не є актуальними або були введені помилково;

2. Категоризація доходів та витрат

- користувачі повинні мати можливість створювати нові категорії для класифікації своїх фінансових операцій;

- система повинна дозволяти призначати категорії кожній транзакції для зручності обліку та аналізу;

3. Генерація звітів

- система повинна забезпечувати генерацію різноманітних фінансових звітів, включаючи щомісячні, квартальні та річні звіти про доходи та витрати;

- застосунок повинен надавати можливість візуалізації даних у вигляді графіків та діаграм для кращого розуміння фінансових тенденцій;

4. Інтеграція з банківськими системами

- система повинна інтегруватися з банківськими системами для автоматичного завантаження та оновлення фінансових транзакцій користувачів;

Нефункціональні вимоги

1. Продуктивність

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 20
Арк.	№ докум.	Підпис	Дата			

- система повинна забезпечувати швидкий час відгуку для всіх основних операцій, таких як додавання, редагування та видалення транзакцій, а також генерація звітів;

- необхідно оптимізувати структуру бази даних для забезпечення ефективного зберігання та швидкого доступу до даних;

2. Безпека

- система повинна використовувати сучасні методи шифрування для захисту персональних даних користувачів від несанкціонованого доступу;

- застосунок повинен забезпечувати надійну систему аутентифікації та авторизації для доступу до персональних даних користувачів;

3. Масштабованість

- система повинна бути здатна обробляти велику кількість одночасних користувачів без зниження продуктивності;

- архітектура додатку повинна передбачати можливість легкого додавання нових функцій та модулів без значних змін у існуючому коді;

4. Кросплатформенність

- вебзастосунок повинен коректно функціонувати на різних типах пристроїв, включаючи десктопи, планшети та смартфони;

- інтерфейс користувача повинен автоматично адаптуватися до розмірів екрану пристрою користувача для забезпечення зручності використання;

Виконання зазначених вимог забезпечить створення функціонального, продуктивного та безпечного вебзастосунку для управління особистими фінансами, що задовольнить потреби користувачів та відповідатиме сучасним стандартам якості.

Обмеження:

1. Підтримка браузерів

- вебзастосунок призначений для роботи в сучасних веббраузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge та Safari. Система може не

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 21
Арк.	№ докум.	Підпис	Дата			

функціонувати коректно в застарілих браузерях або тих, які не підтримують новітні вебтехнології;

- застосунок повинен забезпечувати коректну роботу на мобільних браузерях, але деякі функції можуть бути обмежені через особливості мобільних платформ;

2. Підключення до Інтернету

- для повноцінного функціонування вебзастосунків вимагає постійного підключення до Інтернету. Без доступу до Інтернету деякі функції можуть бути недоступні, включаючи синхронізацію даних та автоматичне завантаження транзакцій з банківських систем;

3. Безпека даних

- система забезпечує захист персональних даних користувачів через використання сучасних методів шифрування. Проте існують обмеження щодо абсолютної безпеки даних в умовах кіберзагроз, тому важливо регулярно оновлювати систему безпеки та слідкувати за новими вразливостями;

4. Обмеження функціоналу

- деякі функції вебзастосунку можуть бути обмежені або недоступні у тестовому середовищі. Повний функціонал доступний лише у виробничій версії системи;

- інтеграція з банківськими системами може бути обмежена через різні політики банків та технічні обмеження API;

Припущення:

1. Базові знання користувачів

- припускається, що користувачі мають базові знання про управління фінансами та здатні самостійно вводити дані про свої фінансові операції;

- вважається, що користувачі володіють базовими навичками користування вебзастосунками та можуть самостійно виконувати основні операції у системі;

2. Коректність даних

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 22
Арк.	№ докум.	Підпис	Дата			

- припускається, що всі дані, введені користувачами, є коректними та достовірними. Відповідальність за правильність введеної інформації покладається на користувачів;

- вважається, що інтеграція з банківськими системами забезпечує коректну та своєчасну синхронізацію фінансових транзакцій без значних затримок або помилок;

3. Технічні обмеження

- вебзастосунок розроблено з урахуванням стандартних можливостей серверного обладнання. Припускається, що сервери мають достатню потужність для обробки запитів користувачів без значних затримок;

- система передбачає можливість масштабування для підтримки зростаючої кількості користувачів. Вважається, що в разі необхідності можна легко додати нові сервери та розширити інфраструктуру;

Ці обмеження та припущення є важливими аспектами при розробці та впровадженні вебзастосунку, оскільки вони визначають рамки його функціональності та ефективності. Їх врахування дозволяє забезпечити належний рівень якості та надійності системи.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 23
	Арк.	№ докум.	Підпис	Дата		

1.4 Висновки

У першому розділі розглянуто створення web-орієнтованої інформаційної системи страхового агентства. Визначено ключові завдання для її ефективного функціонування, включаючи аналіз потреб та викликів. Обрано оптимальний набір технологій: Node для бекенду, HTML, CSS та React для фронтенду, та NoSql db для бази даних, забезпечуючи високу продуктивність, безпеку та зручність інтерфейсу. Розробка системи охоплює аналіз вимог, дизайн, програмування, тестування, та оптимізацію. Висновки підкреслюють актуальність проєкту для страхового агентства, його перспективність у підвищенні якості обслуговування та ефективності бізнес-процесів.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 24
	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Архітектура інформаційної системи

На представленій схемі (Рис. 2.1.) зображено структуру web-орієнтованої інформаційної системи страхового агентства, який складається з двох основних частин: CSS файлів та JavaScript файлів.

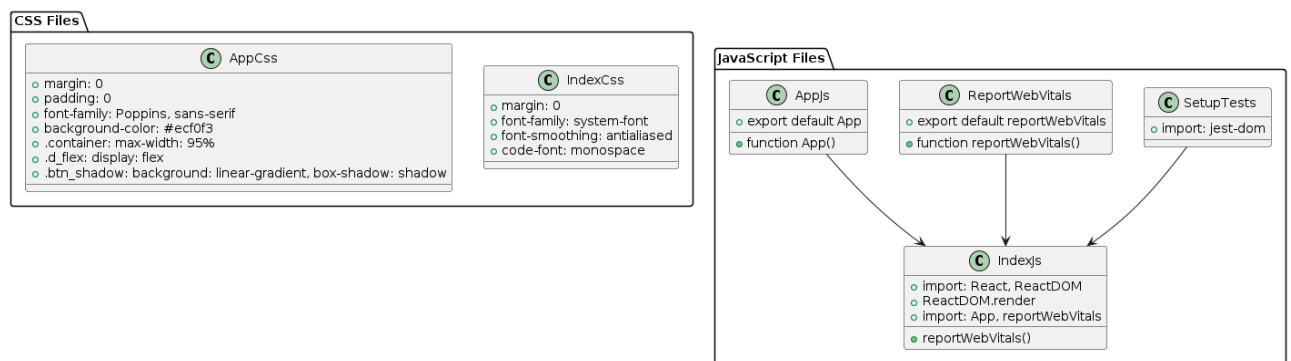


Рисунок 2.1 – Схема структури вебзастосунку

У розділі CSS файлів містяться два основні файли стилів: App.css та index.css. Файл App.css містить глобальні стилі, які застосовуються до всіх компонентів додатку, включаючи відступи, шрифти, кольори фону та стилі для контейнерів та кнопок. Зокрема, використовується шрифт "Poppins", а для кнопок застосовується градієнтний фон та тіні, що надають додатку сучасного вигляду. Файл index.css також містить базові стилі, зокрема визначення шрифтів та згладжування тексту для забезпечення кращої читабельності.

JavaScript файли складаються з кількох ключових компонентів, що забезпечують функціональність додатку. Файл App.js містить основний компонент додатку App, який експортується для використання в інших частинах програми. Він відповідає за відображення основного інтерфейсу користувача. index.js є центральним файлом, який імпортує необхідні бібліотеки та компоненти, такі як React та ReactDOM, і здійснює рендеринг додатку в DOM.

Крім того, він імпортує компонент App та функцію reportWebVitals, яка використовується для збору та аналізу показників продуктивності додатку.

Файл reportWebVitals.js містить функцію reportWebVitals, яка експортується для використання в index.js і відповідає за моніторинг продуктивності додатку, що є важливим для забезпечення швидкодії та оптимізації користувацького досвіду. Файл setupTests.js містить імпорт бібліотеки jest-dom, що використовується для налаштування тестового середовища та проведення тестування компонентів додатку.

Таким чином, схема наочно демонструє зв'язки між основними компонентами вебзастосунку, що забезпечують його функціональність та зовнішній вигляд. Структура файлів відображає модульний підхід до розробки, де кожен компонент відповідає за певну частину додатку, забезпечуючи легкість у підтримці та розширенні функціоналу.

На представленій діаграмі (Рис. 2.2.) відображено процес функціонування web-орієнтованої інформаційної системи страхового агентства з використанням технологій React. Процес починається з завантаження вебсторінки, що ініціює імпорт залежностей, включаючи React, ReactDOM, App та reportWebVitals. Цей етап є критичним для коректного функціонування додатку, оскільки забезпечує необхідні бібліотеки та компоненти для подальшої роботи.

Після імпорту залежностей здійснюється спроба відображення компонента App. Якщо цей процес проходить успішно, виконується функція App, що відповідає за рендеринг основного інтерфейсу користувача. В іншому випадку, у випадку помилки завантаження компонента App, алгоритм передбачає обробку цієї помилки, що дозволяє системі коректно реагувати на непередбачені ситуації та запобігати збої у роботі додатку.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 26
	Арк.	№ докум.	Підпис	Дата		

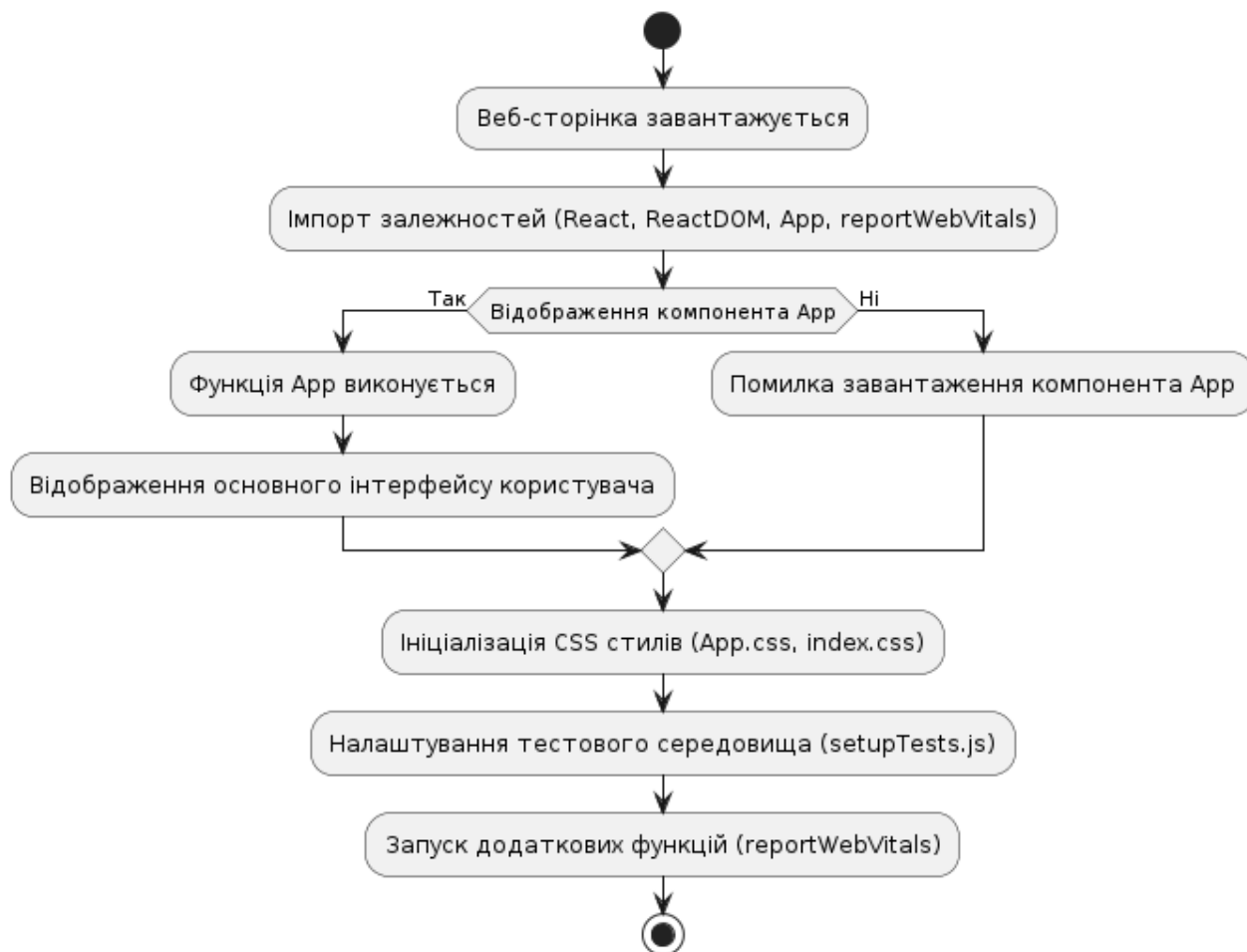


Рисунок 2.2 – Діаграма функціонування вебзастосунку

Після успішного відображення основного інтерфейсу користувача відбувається ініціалізація CSS стилів. У цьому процесі застосовуються стилі з файлів «App.css» та «index.css», що забезпечують єдиний зовнішній вигляд та узгодженість елементів інтерфейсу. Це є важливим етапом для створення приємного та інтуїтивно зрозумілого користувацького досвіду.

Наступним кроком є налаштування тестового середовища за допомогою файлу «setupTests.js». Це дозволяє проводити автоматизоване тестування компонентів додатку, забезпечуючи високу якість та надійність програмного продукту. Використання тестів дозволяє виявляти та виправляти помилки на ранніх етапах розробки, що значно підвищує стабільність системи.

Завершальним етапом є запуск додаткових функцій, таких як «reportWebVitals», які здійснюють моніторинг та збір даних про продуктивність додатку. Ці дані

можуть використовуватися для подальшої оптимізації та покращення роботи системи, забезпечуючи високу швидкодію та ефективність роботи вебзастосунку.

Таким чином, діаграма ілюструє послідовність кроків, що забезпечують повноцінне функціонування вебзастосунку від моменту завантаження сторінки до запуску додаткових функцій та моніторингу продуктивності. Кожен етап є важливим для забезпечення стабільності, продуктивності та зручності використання додатку.

На представленій діаграмі (Рис. 2.3.) відображено структуру бази даних web-орієнтованої інформаційної системи страхового агентства, що складається з чотирьох основних таблиць: Users, Orders, Products та OrderItems. Кожна з цих таблиць має свої атрибути та зв'язки з іншими таблицями, що забезпечують цілісність даних та логіку роботи системи.

Таблиця Users зберігає інформацію про користувачів додатку. Вона містить атрибути: user_id, username, email, password, created_at та updated_at. Первинним ключем цієї таблиці є user_id, що забезпечує унікальну ідентифікацію кожного користувача.

Таблиця Orders містить дані про замовлення, зроблені користувачами. Атрибути цієї таблиці включають: order_id, user_id, order_date, total_amount, status, created_at та updated_at. Первинним ключем є order_id, а user_id є зовнішнім ключем, що встановлює зв'язок з таблицею Users. Цей зв'язок показує, що кожне замовлення належить певному користувачеві, який його розмістив.

Таблиця Products зберігає інформацію про продукти, доступні в системі. Вона включає такі атрибути, як product_id, name, description, price, stock_quantity, created_at та updated_at. Первинним ключем цієї таблиці є product_id, що гарантує унікальність кожного продукту.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 28
	Арк.	№ докум.	Підпис	Дата		

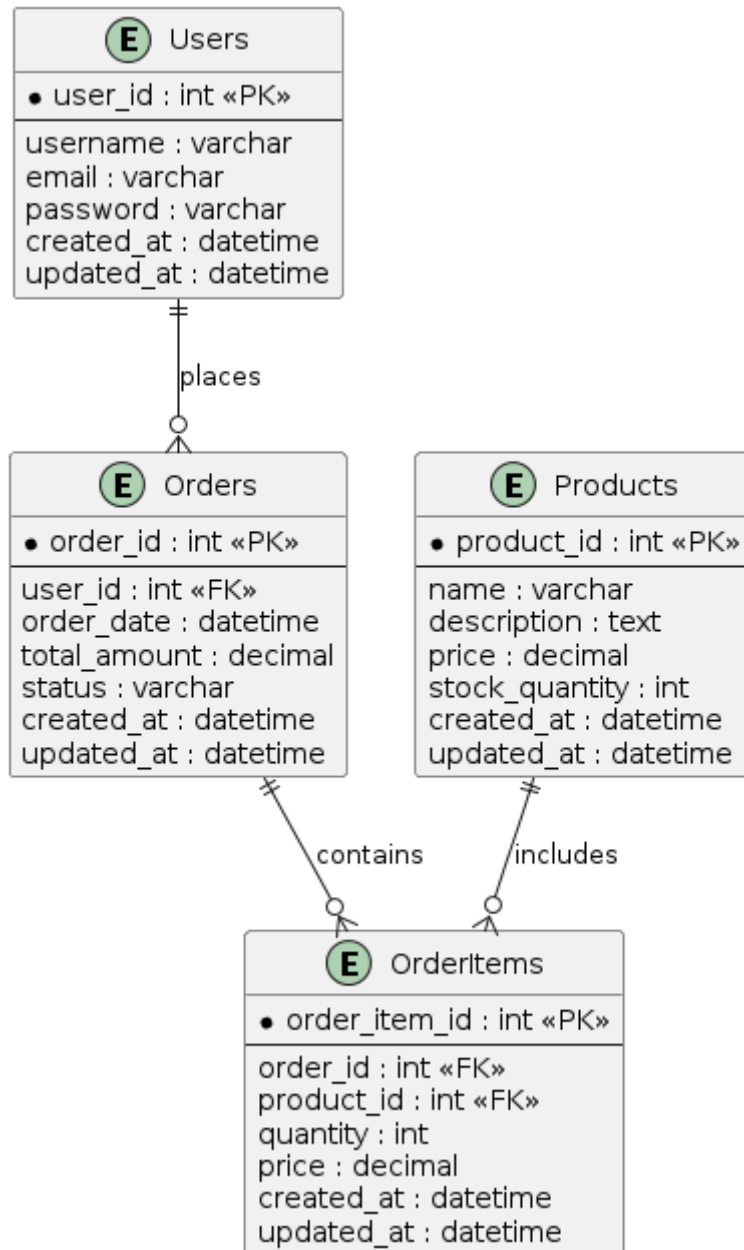


Рисунок 2.3 – Структура бази даних

Таблиця OrderItems деталізує склад кожного замовлення. Вона містить атрибути: order_item_id, order_id, product_id, quantity, price, created_at та updated_at. Первинним ключем є order_item_id. Зовнішні ключі order_id та product_id встановлюють зв'язки з таблицями Orders та Products відповідно. Це дозволяє відстежувати, які продукти були замовлені в кожному конкретному замовленні та в якій кількості.

Зв'язки між таблицями представлені відповідними лініями на діаграмі. Таблиця Orders має зв'язок "багато-до-одного" з таблицею Users, що означає, що кожен користувач може мати багато замовлень. Таблиця OrderItems має зв'язок "багато-до-одного" з таблицею Orders, що вказує на те, що кожне замовлення може містити багато елементів. Аналогічно, таблиця OrderItems має зв'язок "багато-до-одного" з таблицею Products, що означає, що кожен елемент замовлення відповідає певному продукту.

Ця структура бази даних забезпечує ефективне управління даними користувачів, замовлень та продуктів, а також дозволяє легко масштабувати систему та додавати нові функціональні можливості.

У контексті архітектури інформаційної системи, клієнтська частина, або фронтенд, є тією сферою, де відбувається пряме взаємодія з кінцевими користувачами. Основне завдання фронтенду – забезпечити користувачам доступний, інтуїтивно зрозумілий та естетично привабливий інтерфейс, який водночас є функціональним та швидкодіючим.

Структура вебсторінок розробляється на основі HTML – мови розмітки, що визначає кістяк та зміст вебсторінки. За допомогою HTML створюються різноманітні елементи, як-от заголовки, параграфи, списки, таблиці, форми та багато іншого.

CSS використовується для стилізації HTML-структури, додаючи до неї візуальну атрактивність. За допомогою CSS можна регулювати шрифти, кольори, відстані, фони елементів, а також створювати складніші ефекти, такі як анімації, трансформації та переходи.

JavaScript використовується для надання інтерактивності вебсторінці. Він дозволяє реалізувати динамічні зміни контенту без перезавантаження сторінки, обробку подій користувача, а також створення більш складних користувацьких сценаріїв, таких як робота з формами та взаємодія з сервером.

React, як сучасна бібліотека для розробки інтерфейсів, істотно підвищує ефективність процесу розробки, оскільки вона вносить компонентний підхід у

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 30
Арк.	№ докум.	Підпис	Дата			

структурування UI. Завдяки цьому, UI може бути розбитий на перевикористовувані компоненти, що спрощує управління станом, полегшує розширення та підтримку додатків.

Webpack є модульним збірником для JavaScript-додатків, який управляє залежностями та оптимізує завантаження ресурсів. Babel – це транспілятор, який перетворює сучасний JavaScript у код, сумісний з більшістю веббраузерів, забезпечуючи розробникам можливість використання останніх особливостей мови без ризику несумісності.

У сукупності, ці технології формують міцну основу для створення швидкодіючих та відзивчивих односторінкових вебзастосунків (SPA), які пропонують користувачам високоякісний та безшовний досвід, необхідний для сучасного цифрового середовища. Використання SPA знижує час завантаження контенту, мінімізує затримки в інтерфейсі користувача та покращує загальну взаємодію з додатком, що сприятливо позначається на задоволеності користувачів та ефективності їх взаємодії зі страховою компанією.

У структурі сучасних інформаційних систем серверна частина займає ключову позицію, оскільки саме вона відповідає за обробку та управління даними, здійснює комунікацію з базами даних і проміжне програмне забезпечення, а також забезпечує обробку і виконання запитів клієнтської частини. Розвиток вебтехнологій дозволив використовувати JavaScript не тільки для фронтенду, а й на стороні сервера, що розширює можливості і підвищує ефективність розробки. В цьому контексті Node.js як серверне середовище виконання JavaScript стає вибором для розробки бекенду завдяки своїй продуктивності, масштабованості та гнучкості.

Node.js пропонує модель неблокуючого вводу-виводу та асинхронні операції, що дозволяє системі обробляти велику кількість запитів ефективно і з мінімальним затриманням. Це створює оптимальні умови для розробки високонавантажених вебзастосунків, таких як API та сервіси реального часу.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 31
Арк.	№ докум.	Підпис	Дата			

Бекенд, створений на базі Node.js, легко інтегрується з різноманітними системами управління базами даних через обширну бібліотеку драйверів та ORM інструментів, що дозволяє забезпечити надійне, гнучке та безпечне зберігання даних. Розробники можуть вибирати серед популярних СУБД як MongoDB для документ-орієнтованих даних, так і MySQL або PostgreSQL для реляційних баз даних, в залежності від потреб проєкту.

Наявність великої кількості модулів та інструментів у екосистемі Node.js, доступних через npm, значно спрощує процес розробки, оскільки дозволяє використовувати готові рішення для стандартних задач, інтегрувати зовнішні API та служби, а також застосовувати передові практики без необхідності "винаходити колесо".

Узагальнюючи, бекенд на базі Node.js в інформаційних системах страхових агентств відкриває широкі можливості для створення потужних, гнучких та масштабованих вебзастосунків. Він не лише забезпечує високу продуктивність обробки даних і запитів, але й сприяє оперативному впровадженню нових бізнес-функцій та послуг, підвищуючи здатність системи швидко адаптуватися до змін на ринку страхових послуг.

Прийняття архітектурного підходу Модель-Представлення-Контролер (MVC) (Рис. 2.4.) при проектуванні серверної частини на базі Node.js дозволяє забезпечити чітку організацію та відділення логіки додатку від його візуальної презентації. Даний патерн сприяє створенню структурованих і ефективно управляємих вебзастосунків.

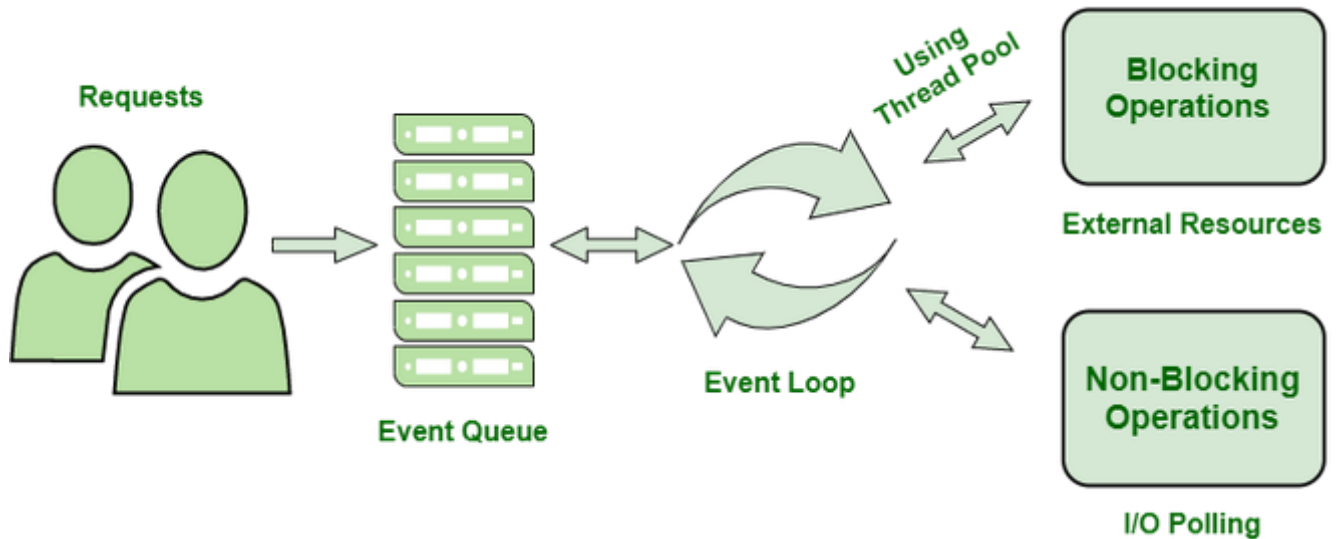


Рисунок 2.4 – Архітектура Node

Моделі у такій архітектурі є фундаментом, що визначає логічну структуру даних, а також їх взаємозв'язки всередині бази даних. Використовуючи ORM-інструменти, розробники мають змогу абстрагувати та маніпулювати даними бази в термінах об'єктів і класів, замість прямого складання SQL-запитів, що суттєво спрощує роботу з даними та зменшує кількість потенційних помилок.

Представлення у Node.js відповідають за генерацію користувацьких інтерфейсів, трансформуючи дані, отримані від моделей, у зручний для сприйняття формат. Використання шаблонів і шаблонізаторів, таких як EJS або Handlebars, дозволяє розробникам створювати динамічні вебсторінки, які можуть адаптуватися та змінюватися залежно від поточних даних і дій користувача.

Контролери виконують роль оркестраторів в MVC архітектурі, приймаючи запити від користувачів, викликаючи відповідні моделі для отримання або оновлення даних, і вибираючи відповідні представлення для відображення результату. Ця роль включає в себе і визначення логіки обробки даних, забезпечуючи таким чином поділ та організацію програмного коду.

Архітектура MVC, реалізована з використанням Node.js, сприяє створенню коду, що є більш чистим, організованим та легшим для тестування та підтримки. Вона дозволяє розробникам ефективно використовувати ресурси, спрямовувати та

управляти різноманітними аспектами бізнес-логіки, а також підвищувати масштабованість та продуктивність додатків. Використовуючи надійні бази даних та екосистему Node.js, MVC архітектура надає міцну основу для розвитку великих та вимогливих до продуктивності вебзастосунків, відповідаючи вимогам до швидкості обробки та гнучкості сучасних даних.

2.1 Функціональні вимоги до інформаційної системи

Інформаційна система страхової компанії повинна забезпечувати комплексний підхід до управління страховими продуктами та послугами, а також гарантувати високий рівень взаємодії з користувачами. Система має включати:

- повноцінну інтеграцію з внутрішніми ресурсами компанії для забезпечення актуальності інформації, включаючи останні новини та оновлення в сфері страхування.
- зручні інструменти для опису, класифікації та управління страховими продуктами, що дозволяють користувачам легко орієнтуватися серед пропозицій;
- модуль оцінки вартості страхових полісів, який забезпечує прозорість та дозволяє користувачам самостійно розрахувати вартість послуг;
- систему відгуків, яка дозволяє збирати та аналізувати зворотній зв'язок від клієнтів, сприяючи покращенню якості обслуговування;
- функціонал контактної інформації та зворотного зв'язку для ефективної комунікації між користувачами та страховою компанією;
- засоби для забезпечення конфіденційності та захисту персональних даних користувачів, включаючи шифрування даних та інші технології безпеки;

Реалізація цих функціональних вимог забезпечить ефективну взаємодію користувачів зі страховою системою, підвищить їх лояльність та довіру до компанії, а також сприятиме оперативному управлінню ризиками та страховими випадками. Система повинна бути гнучкою для адаптації до змін у страховому ринку та здатною масштабуватися відповідно до зростаючих потреб бізнесу.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 34
Арк.	№ докум.	Підпис	Дата			

2.3 Нефункціональні вимоги до інформаційної системи

Нефункціональні вимоги до інформаційної системи страхового агентства охоплюють важливі аспекти, які безпосередньо не відносяться до конкретних операцій системи, але є вирішальними для загального її функціонування, надійності, та сприйняття користувачами. Вони включають, але не обмежуються, наступними критеріями:

Система має демонструвати високу надійність, забезпечуючи безперебійну роботу та мінімальну кількість збоїв. Це стосується як внутрішніх операцій, так і взаємодії з користувачами, забезпечуючи неперервне надання послуг незалежно від обставин.

Безпека є важливою вимогою, яка стосується захисту даних користувачів від несанкціонованого доступу та інших потенційних кіберзагроз. Система має включати сучасні заходи шифрування, аутентифікації, та інші технології безпеки.

Продуктивність системи має забезпечувати швидке та ефективно оброблення запитів користувачів, здатність витримувати високе навантаження під час пікових періодів без втрати швидкості відгуку.

Масштабованість вказує на здатність системи адаптуватися до росту кількості користувачів, транзакцій, та обсягів даних, розширюючись без потреби в радикальних змінах архітектури чи впровадження нового обладнання.

Доступність є критичною для користувачів, які очікують доступу до системи в будь-який час та з будь-якої точки світу, на різних пристроях, що вимагає від системи стабільної роботи та оптимальної адаптації до різних умов використання.

Сумісність гарантує, що інформаційна система може інтегруватися та працювати з іншими існуючими системами та платформами, які використовуються у страховій індустрії, забезпечуючи синергію та оптимізацію робочих процесів.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 35
Арк.	№ докум.	Підпис	Дата			

Загалом, нефункціональні вимоги до інформаційної системи страхового агентства служать фундаментом для створення продукту, який є не тільки ефективним та функціональним, але також надійним, безпечним і зручним для користувачів. Забезпечення стабільної та неперервної роботи системи, захисту даних та особистої інформації користувачів, швидкості та ефективності обробки запитів, масштабованості для підтримки зростаючої кількості користувачів і даних, гарантованої доступності та сумісності з іншими системами — все це визначає нефункціональні параметри, які впливають на загальне враження користувача від роботи з системою та її прийняття на ринку.

2.4 Користувацькі вимоги до інформаційної системи страхового агентства

Розробка інформаційної системи страхового агентства передбачає тісну взаємодію з кінцевими користувачами, що вимагає глибокого розуміння їхніх потреб і переваг. Важливим аспектом є створення інтуїтивно зрозумілого інтерфейсу, який забезпечує легкий доступ до необхідних функцій і сервісів без потреби в глибоких технічних знаннях. Це означає, що навігація по системі має бути логічною та простою, з чітко визначеними шляхами до кожної операції, від інформації про компанію до виконання специфічних запитів, таких як оформлення поліса чи подання заявки на виплату. Персоналізація користувацького досвіду є ще однією важливою вимогою, яка дозволяє користувачам налаштувати систему під власні потреби. Це може включати в себе налаштування повідомлень, управління персональними даними та інтерфейсом, а також кастомізацію звітів та аналітики.

Доступність системи з різних пристроїв, що охоплює широкий спектр гаджетів, від мобільних телефонів до настільних комп'ютерів, є необхідною для забезпечення гнучкості користувачів у виборі зручного способу взаємодії з сервісом. Це вимагає від системи високої адаптивності та респонсивності дизайну, забезпечуючи користувачам зручне відображення та управління на будь-

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 36
Арк.	№ докум.	Підпис	Дата			

якому пристрої. Підтримка багатомовності дозволяє зробити систему доступною для міжнародного кола користувачів, адаптуючи контент та інтерфейс до їхніх мовних уподобань, що є ключовим аспектом для глобалізації бізнесу. Швидкість та ефективність системи мають велике значення для забезпечення високого рівня задоволення користувачів. Час відгуку на дії користувача має бути мінімізований, щоб забезпечити негайну відповідь на запити та ефективність виконання робочих процесів. Легкий доступ до підтримки та послуг консультування є необхідним для забезпечення високої якості обслуговування. Користувачі повинні мати можливість з легкістю знаходити інформацію та отримувати допомогу через різні канали, включаючи онлайн-чати, електронну пошту, телефонні дзвінки або навіть через вбудовані в систему інструменти самопомоги.

Захист персональних даних є ще однією важливою користувацькою вимогою. Інформаційна система має інкорпорувати передові методи шифрування та забезпечення конфіденційності, щоб відповідати міжнародним стандартам та законодавчим вимогам з охорони даних.

Окрім того, система повинна бути стабільною та надійною, забезпечуючи неперервну роботу та мінімальні перерви в обслуговуванні. Впровадження ефективних механізмів моніторингу та резервного копіювання є критичними для попередження втрат даних та забезпечення швидкого відновлення системи після будь-яких збоїв.

Наостанок, для гарантування відповідності системи потребам різних груп користувачів, слід розглянути можливість налаштування індивідуальних робочих просторів або персоналізованих дашбордів, які відображатимуть найважливішу інформацію відповідно до ролі та функцій кожного користувача в системі.

Увага до цих користувацьких вимог та їхня ретельна реалізація сприятиме створенню системи, яка не тільки задовольняє технічні аспекти функціонування страхового агентства, але й вибудовує позитивні відносини з клієнтами та забезпечує їхню лояльність і довіру.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 37
Арк.	№ докум.	Підпис	Дата			

2.5 Висновки

Завершальний аналіз проектування інформаційної системи для страхового агентства дозволяє сформулювати кілька ключових висновків. У процесі детального вивчення функціональних та нефункціональних вимог було виявлено, що успіх системи залежить від її здатності гармонійно поєднувати високу ефективність управління процесами страхування з надійністю, безпекою та доступністю для користувачів. Важливу роль відіграє інтуїтивно зрозумілий інтерфейс, що сприяє кращій взаємодії користувачів з системою та підвищує їх задоволеність сервісом.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						38
	Арк.	№ докум.	Підпис	Дата		

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСТОСУНКУ

3.1 Огляд системи та вибір технологій

Інформаційна система страхування створена на основі найновіших технологій веброзробки. Система спрощує управління страховими продуктами та послугами, надаючи користувачам зручний онлайн-доступ до різноманітних страхових полісів, і дозволяє страховій компанії вести ефективну взаємодію з клієнтами.

Основною функціональністю цього проекту є відображення детальної інформації про страхові послуги, а також надання користувачам можливості здійснення заявок на страхування та перегляду актуальних новин у сфері страхування. Система також забезпечує доступ до інформаційних ресурсів та аналітичних даних про страховий ринок.

Для розробки проекту використовуються такі технології:

– бібліотека React для побудови користувацького інтерфейсу, яка дозволяє створювати швидкі та відгукові вебсторінки. React використовується для реалізації динамічної взаємодії на стороні клієнта, забезпечуючи гладке та інтуїтивно зрозумілий користувацький досвід;

– подієво-орієнтоване Node.js середовище виконання для JavaScript, що дозволяє створювати швидкодіючі та масштабовані мережеві додатки. Node.js використовується для побудови серверної частини системи, яка відповідає за обробку даних та інтеграцію з базами даних;

– легковагий та гнучкий фреймворк Express.js для Node.js, який використовується для створення вебзастосунків та API з простими та зрозумілими маршрутами та обробкою запитів;

– використання NoSQL Web API дозволяє системі легко адаптуватися до різних форматів даних і є ідеальним рішенням для роботи з великими обсягами неупорядкованих або структурованих даних, забезпечуючи при цьому високу продуктивність та автоматичне масштабування;

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 39
Арк.	№ докум.	Підпис	Дата			

Цей набір технологій надає системі необхідну гнучкість та масштабованість, водночас забезпечуючи високий рівень зручності та доступності для користувачів. Вебінтерфейс системи розроблено з врахуванням кращих практик UX/UI дизайну, що дозволяє забезпечити відмінну навігацію та взаємодію з системою.

Для забезпечення зворотного зв'язку з користувачами система має інтегрований модуль зв'язку, який дозволяє користувачам легко звернутися до служби підтримки. Система також оснащена інструментами для збору статистики та аналізу поведінки користувачів, що надає цінну інформацію для подальшого розвитку та вдосконалення послуг.

3.2 Деталізація бази даних та загальна архітектура.

3.2.1 Архітектура та деталізація бази даних

Система управління страхуванням, яка була розроблена, використовує сучасні вебтехнології та заснована на принципах взаємодії через WEB API. Всі дані обробляються та передаються між клієнтом і сервером у форматі JSON через зазначені API, що забезпечує швидкість та гнучкість системи.

Структура даних в інформаційній системі управління страхуванням відмовляється від традиційних реляційних таблиць на користь більш гнучкої та масштабованої NoSQL архітектури. Замість статичних схем баз даних, система використовує документо-орієнтовану модель, яка спрощує внесення змін, розширення та інтеграцію з різноманітними джерелами даних.

Схема на рисунку (Рис. 3.1) ілюструє схему взаємодії з WEB API, де кожний ресурс, такий як поліси або користувачі, представлений у вигляді окремого документа. Ці документи зберігаються в колекціях, які можна легко змінювати, фільтрувати та масштабувати без потреби в передбаченні відносин між даними.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						40
Арк.	№ докум.	Підпис	Дата			

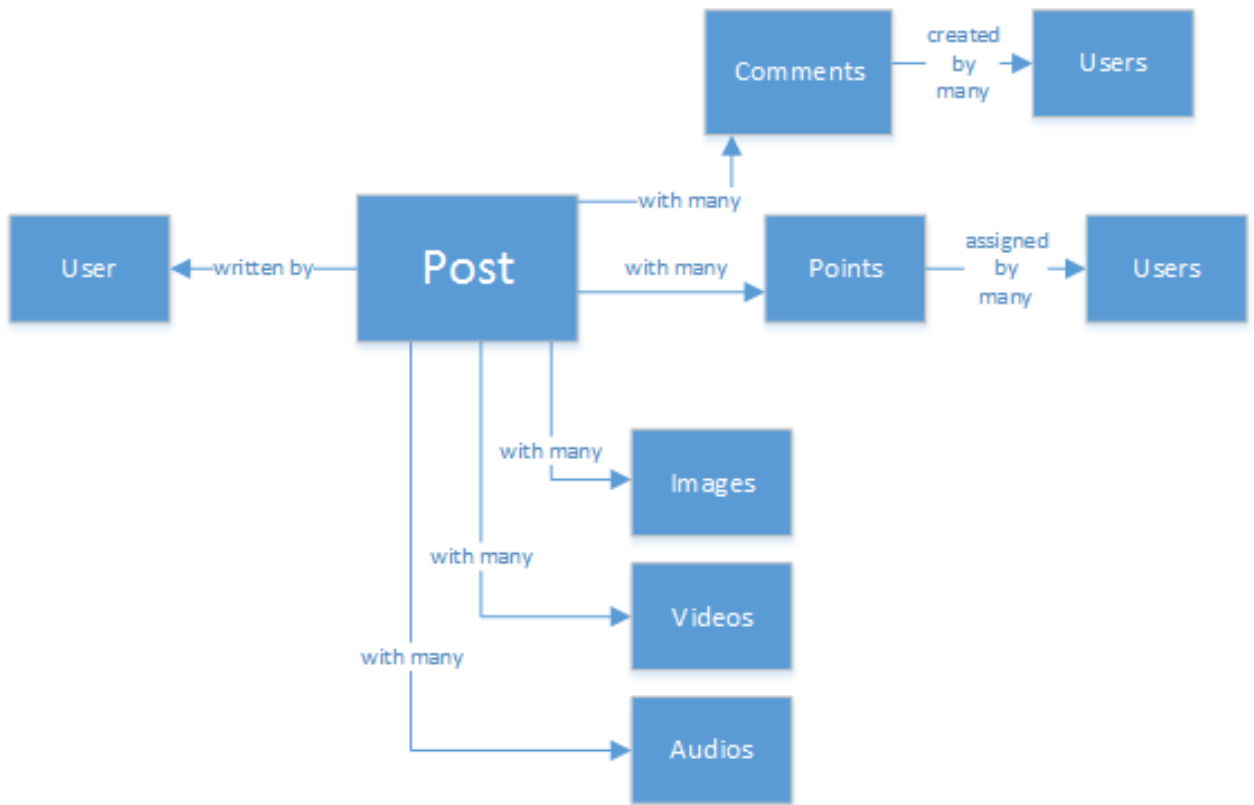


Рисунок 3.1 – WEB API схема взаємодії бази даних

Таблиця користувачів замінена колекцією документів, де кожен документ містить повну інформацію про користувача, включно з даними для входу, персональною інформацією та налаштуваннями. Це усуває потребу в окремих таблицях для авторизації та аутентифікації, оскільки всі ці функції тепер вбудовані безпосередньо у структуру даних.

Щодо страхових полісів, кожний поліс тепер є самостійним документом, що містить всі відомості про умови, вартість та терміни дії. Це дозволяє системі швидко обробляти заявки, оновлювати політику страхування та надавати користувачам актуальну інформацію.

Вебзапити до API обслуговують динамічні запити від користувачів, такі як створення нового полісу або оновлення існуючого. Асоціації між документами забезпечуються за допомогою вказівників на інші документи, що дозволяє створювати складні взаємозв'язки між даними, як-от зв'язок користувача з його страховими полісами.

Така організація даних значно спрощує розробку та майбутнє масштабування системи, оскільки дозволяє додавати, видаляти або оновлювати контент без необхідності зміни структури бази даних або втручання в бізнес-логіку системи.

У системі імплементовано сучасний фронтенд, створений з використанням React, що надає користувачам інтуїтивно зрозумілий та зручний інтерфейс. Інтеграція з бекендом здійснюється через вебзапити до API, що дозволяє відмовитися від традиційного серверного рендерингу сторінок і зробити загрузку інтерфейсу максимально швидкою та ефективною.

Таким чином, обрані технології та архітектурні рішення дозволяють системі бути сучасною, надійною та зручною для кінцевих користувачів, в той же час забезпечуючи легкість у підтримці та масштабуванні з боку розробників.

3.2.2 Використання архітектурного патерну Модель-Вид-Шаблон (MVT) у Node

Використання Node.js та Express.js у якості серверної платформи та фреймворку надає низку переваг для розробки вебзастосунків. Node.js є потужним середовищем виконання коду на стороні сервера, що використовує неблокуючу модель вводу/виводу, дозволяючи створювати швидкі та масштабовані мережеві додатки. Express.js, будучи мінімалістичним та гнучким фреймворком для додатків Node.js, забезпечує набір потужних функцій для веброзробки, включаючи маршрутизацію запитів, обробку помилок, середнє програмне забезпечення (middleware) та більше.

У системі управління страхуванням, Node.js та Express.js можуть бути використані для створення RESTful API, що слугуватиме центральним інтерфейсом для зв'язку між клієнтською частиною (фронтендом) та сервером. RESTful API дозволяє легко створювати, читати, оновлювати та видаляти ресурси (CRUD операції), які відображають бізнес-логіку страхових полісів та клієнтських відносин.

```

1  import React, { useState } from 'react'
2
3
4  const Card = (props) => {
5
6      const [modal, setModal] = useState(false)
7      const toggleModal = () => {
8          setModal(!modal)
9      }
10     return (
11         <div className='box btn_shadow'>
12             <div className='img'>
13                 <img src={props.image} alt="" onClick={toggleModal} />
14             </div>
15             <div className='category_d_flex'>
16                 <span onClick={toggleModal}>{props.category}</span>
17                 <label>
18                     <i className='far fa-heart'></i>
19                     {props.totalLike}
20                 </label>
21             </div>
22             <div className='title'>
23                 <h2 onClick={toggleModal}>{props.title}</h2>
24                 <a href='#popup' className='arrow' onClick={toggleModal}>
25                     <i className='fas fa-arrow-right'></i>
26                 </a>
27             </div>
28         </div>
29     )
30 }

```

Рисунок 3.3 – Приклад розробленого шаблону та компонентів

У такій архітектурі, Node.js відіграє роль виконання серверного коду, Express.js обробляє HTTP запити та надає відповіді, а фронтенд споживає ці дані через API виклики, створюючи багатий та інтерактивний користувацький досвід.

```

1  import React from "react"
2  import Header from "./Component/Header/Header"
3  import "./App.css"
4  import Home from "./Component/Hero/Home"
5  import Features from "./Component/Features/Features"
6  import Portfolio from "./Component/Portfolio/Portfolio"
7  import Resume from "./Component/Resume/Resume"
8  import Testimonial from "./Component/Testimonial/Testimonial"
9  import Blog from "./Component/Blog/Blog"
10 import Contact from "./Component/Contact/Contact"
11 import Footer from "./Component/Footer"
12
13
14 const App = () => {
15     return (
16         <div>
17             <Header />
18             <Home />
19             <Features />
20             <Portfolio />
21             <Resume />
22             <Testimonial />
23             <Blog />
24             <Contact />
25             <Footer />
26         </div>
27     )
28 }

```

Рисунок 3.4 – Встановлення маршрутизації

Завдяки цьому підходу, система стає більш масштабованою та простою у підтримці, оскільки зміни в одній частині системи (наприклад, додавання нових функцій у API) не вимагають змін у інших частинах (наприклад, у фронтенді). Водночас, система забезпечує швидку обробку запитів та високу продуктивність, які є критично важливими для систем управління страхуванням, де відповіді у реальному часі можуть мати значний вплив на бізнес-процеси.

3.3 Огляд ключових модулів системи

Система управління страхуванням, як складний вебзастосунок, складається з низки взаємопов'язаних модулів, кожен з яких виконує специфічні завдання і забезпечує необхідні сервіси як для клієнтів, так і для адміністрації. Централізоване управління цими модулями дозволяє системі функціонувати як єдиний механізм, забезпечуючи плавне використання ресурсів та оптимальний досвід для користувачів. Для забезпечення інтуїтивно зрозумілого та зручного інтерфейсу, розробка модулів ведеться з урахуванням сучасних принципів UX/UI дизайну, а також найкращих практик програмування.

Модуль Home є головним інтерфейсом користувача і відіграє ключову роль у взаємодії з відвідувачами сайту. Це перше, що вони бачать, відвідуючи систему управління страхуванням, тому важливо, щоб дизайн та функціональність були виконані на високому рівні.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						45
	Арк.	№ докум.	Підпис	Дата		

```
Home.js M X
src > Component > Hero > Home.js > justifyContent
1 import React from 'react'
2 import './Home.css'
3 import hero from "../arch-insurance-logo-vector.svg"
4 import skill1 from "../pic/skill1.png"
5 import skill2 from "../pic/skill2.png"
6 import skill3 from "../pic/skill3.png"
7 import Typewriter from "typewriter-effect";
8
9 const Home = () => {
10   return (
11     <
12       <section className='hero' id='home'>
13         <div className='container f_flex top'>
14           <div className='left top'>
15             <h3></h3>
16             <h1>Ваш надійний помічник у світі <span>страхування</span></h1>
17             <h2 className='effect'>
18               <span className='a_effect'>Розкрийте потенціал </span>
19               <span className='options_span'>
20                 <Typewriter options={{ strings: ["інноваційних сервісів", "страхових полісів"], autoStart: true, loop: true }} />
21             </span>
22           </h2>
23     </div>
```

Рисунок 3.5 – Реалізація модулю home

Основна структура модуля Home включає:

- секцію-хіро (Hero Section), яка запрошує користувачів з динамічним заголовком, що використовує Typewriter-effect для підкреслення інноваційних сервісів та страхових полісів, що надає сайту сучасного і динамічного вигляду;
- посилання на соціальні профілі компанії забезпечують користувачам зручний доступ до спільноти та останніх оновлень;
- інтерактивні кнопки з іконками для корисних ресурсів та інструментів, що стимулюють користувачів до взаємодії;
- зображення-лого компанії з візуальними елементами, яке відображає бренд і привертає увагу до корпоративного іміджу;

Цей модуль Home є серцем інформаційної системи, що об'єднує візуальний дизайн та користувацький досвід, забезпечуючи легкий та ефективний спосіб для користувачів взаємодіяти з основними послугами та функціоналом сайту.

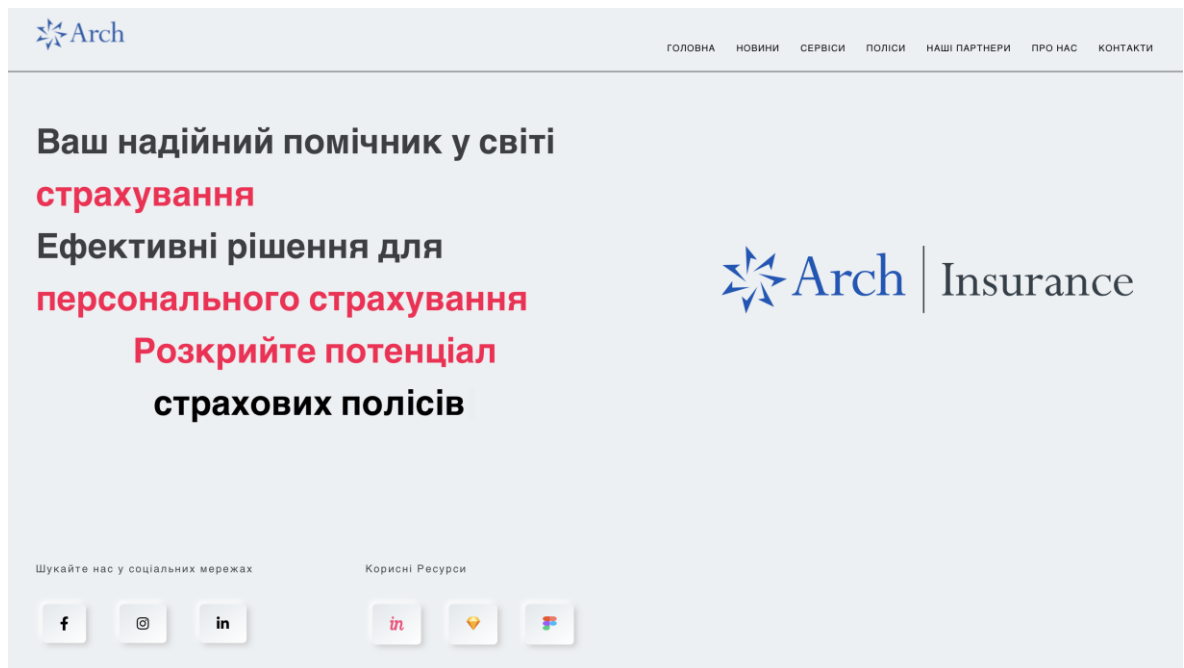


Рисунок 3.6 – Реалізація вітального модулю

Використання CSS для стилізації та React для функціональності робить цей модуль гнучким та легким у масштабуванні, що є важливим аспектом для подальшого розвитку та оновлення системи. Модуль містить інтерактивні елементи, такі як анімовані типографічні ефекти, які акцентують увагу на інноваційних сервісах та страхових продуктах. Водночас він спрямовує користувачів до соціальних мереж компанії та корисних ресурсів, забезпечуючи додаткові шляхи для взаємодії та поглиблення залучення.

Модуль Features.js в інформаційній системі виконує роль вітрини актуальних новин зі сфери страхування. Цей компонент важливий для представлення основної інформації про страхові новини.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 47
Арк.	№ докум.	Підпис	Дата			

```
Features.js M X
src > Component > Features > Features.js > Features
1 import React from 'react'
2 import './Features.css'
3 import Card from './Card'
4 import data from './FeaturesApi'
5
6 const Features = () => {
7   return (
8     <div className="features">
9       <div className="container">
10        <div className="heading">
11          <h4>Огляд останніх новин </h4>
12          <h1>Актуальні новини страхування</h1>
13        </div>
14        <div className="content grid">
15          {data.map((value, index) => {
16            return <Card key={index} image={value.image} title={value.title} desc={value.desc} />
17          })}
18        </div>
19      </div>
20    </div>
21  )
22 }
23
24
25
```

Рисунок 3.7 – Модуль Features

Структура Features.js базується на компонентному підході React і використовує декларативну візуалізацію даних. Всередині компоненту реалізовано відображення колекції елементів, які беруться з даних, отриманих через FeaturesApi.js. Цей API-модуль надає набір даних, який містить інформацію про різні аспекти послуг компанії, включно з описом та зображеннями.

Для кожної новини створюється окрема картка — компонент Card.js, який рендерить інформацію в структурованому і візуально привабливому форматі. Це робить інформацію доступною та зрозумілою для кінцевого користувача, сприяє кращій взаємодії користувачів із системою, а також підвищує шанси на залучення клієнтів до послуг, які вони шукають.

Використання map() для перебору масиву даних з FeaturesApi.js у Features.js забезпечує гнучкість та легкість у додаванні чи оновленні інформації про послуги без необхідності вручну змінювати компоненти. Це демонструє силу компонентної архітектури React, де можна легко масштабувати та управляти контентом.

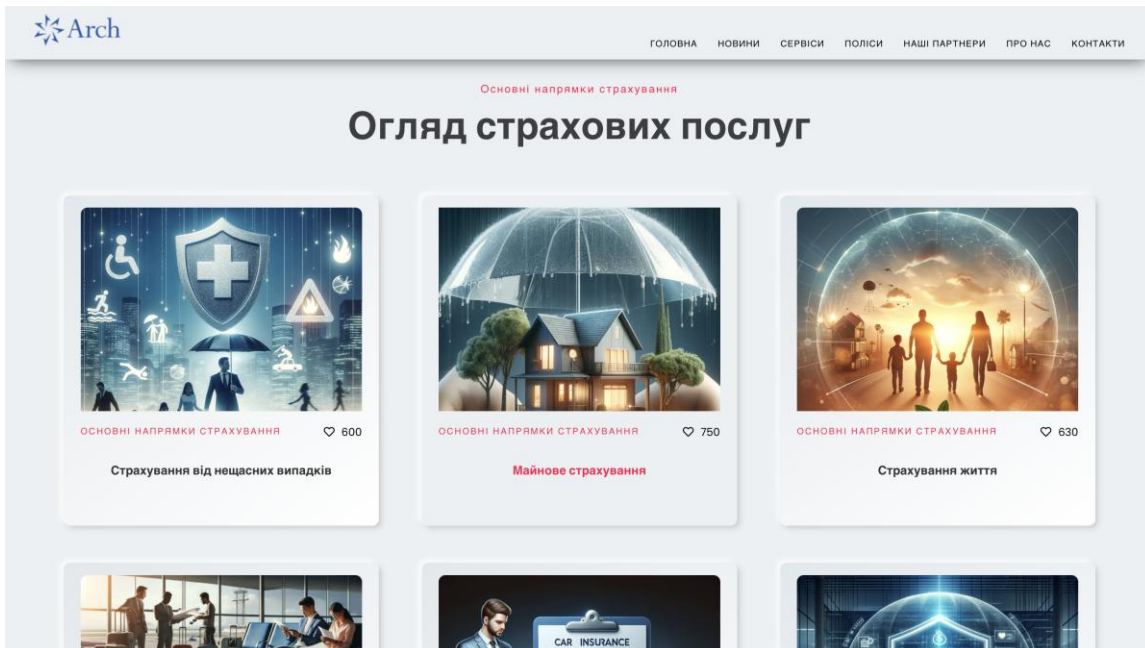


Рисунок 3.9 – Реалізація модулю

Основна функція цього компонента полягає в ефективному представленні інформації, зручному перегляді та легкості доступу до деталей кожного типу страхування, дозволяючи користувачам робити обґрунтований вибір страхових продуктів. Компонент також підтримує інтерактивність, таку як перехід до повного опису страхового продукту або послуги за допомогою натискання на карточку, що забезпечує кращий користувацький досвід.

Компонент Resume являє собою ключову частину web-орієнтованої інформаційної системи страхової агенції та слугує центральним елементом для відображення і вибору страхових полісів користувачами. Завдяки інтеграції сучасних технологій та інтерактивності, він значно підвищує якість користувацького досвіду та ефективність подання інформації.

```
Resume.js × Card.js M
src > Component > Resume > Resume.js > [0] Resume
1 import React from 'react';
2 import './Resume.css';
3 import ResumeApi from './ResumeApi';
4 import Card from './Card';
5
6 const Resume = () => {
7
8   const handleCardClick = (link) => {
9
10    if (link) {
11      window.location.href = link;
12    }
13  };
14
15  return (
16    < >
17    <section className='Resume' id='resume'>
18      <div className='container top'>
19        <div className='heading text-center'>
20          <h4>Розрахунок Полісів</h4>
21          <h1>Розрахуйте ваше страхування за допомогою передових технологій Страхування</h1>
22        </div>
23        <div className='content-section mtop d_flex'>
24          <div className='left'>
25            <div className='heading'>
26              <h4>Популярні страхові поліси</h4>
27              <h1>Обирайте найкраще</h1>
28            </div>
29            <div className='content'>
30              <div className='content'>
31                {ResumeApi.map((value, id) => {
32                  if (value.category === "education") {
33                    return (
```

Рисунок 3.10 – Модуль Resume

Компонент складається з двох основних розділів: "Популярні страхові поліси", де висвітлюються добре знані та часто вибрані страхові продукти, та "Страхові новинки", що представляють останні пропозиції на ринку страхування. Таке розподілення дозволяє користувачам легко навігувати між традиційними та інноваційними страховими рішеннями.

Функція `handleCardClick`, яка забезпечує реакцію на клік по карточці, відіграє важливу роль у навігації користувача, оскільки здійснює перехід на відповідну сторінку з більш детальною інформацією про обраний страховий продукт. Це не тільки спрощує процес знаходження необхідної інформації, але й стимулює користувача до дії, що може позитивно вплинути на конверсію відвідувань у продажі.

Використання масиву даних `ResumeApi` дозволяє системі бути гнучкою та легко адаптованою до змін на ринку страхування, оскільки адміністратори можуть

оновлювати та додавати нові пропозиції без прямого втручання у код. Це також сприяє легкості обслуговування та зменшенню часу, необхідного для оновлення інформації на сайті.

В цілому, компонент Resume вносить значний вклад у підвищення ефективності інформаційної системи страхової агенції, оскільки він відповідає сучасним вимогам до вебресурсів, забезпечуючи високий рівень користувацького сервісу, інтуїтивно зрозумілий інтерфейс та швидкий доступ до необхідної інформації.

3.4 Тестування ефективності системи

Тестування ефективності системи — ключовий компонент гарантування якості веборієнтованої інформаційної системи страхового агентства. Цей процес дозволяє не тільки підтвердити відповідність розробленого програмного продукту встановленим вимогам, але й оцінити його ефективність, надійність та зручність для кінцевих користувачів.

Методологія тестування включає кілька видів: функціональне тестування, яке перевіряє правильність роботи окремих функцій системи; навантажувальне тестування, що вимірює стабільність і швидкість реакції системи під час високих навантажень; тестування безпеки для перевірки захищеності даних користувачів; і юзабіліті тестування для оцінки зручності інтерфейсу.

Для проведення тестувань ми використовуємо такі інструменти, як Selenium для автоматизації браузерних тестів, JMeter для симуляції великого обсягу одночасних запитів до сервера та Burp Suite для аналізу безпеки вебзастосунків. Застосування цих інструментів дозволяє нам систематично і точно перевіряти кожен аспект системи.

Процес тестування розпочинається з розробки детальних тест-кейсів, які охоплюють всі функціональні та нефункціональні вимоги до системи. Тест-кейси плануються відповідно до пріоритетності функцій та ризику, який може

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 52
Арк.	№ докум.	Підпис	Дата			

виникнути в результаті їх некоректної роботи. Всі тестові сценарії документуються і включають очікувані результати, що дозволяє автоматично верифікувати відповідність системи встановленим критеріям.

Збір та аналіз результатів тестування є вирішальним етапом у процесі оцінки ефективності інформаційної системи страхового агентства. Використання детальних метрик дозволяє не тільки виміряти кількісні показники системи, але й забезпечує глибоке розуміння якості виконання програмного забезпечення. Основні метрики, які були використані під час тестування, включають час відгуку системи, кількість оброблених транзакцій на секунду, точність виконання запитів до бази даних, використання ресурсів і доступність системи.

Для моніторингу вебтрафіку використовувався Google Analytics, що дозволило отримати точні дані про відвідування сторінок системи користувачами. Для аналізу продуктивності бази даних використовувався інструмент Database Performance Analyzer, який надав детальну інформацію про час виконання запитів та ефективність використання ресурсів сервера.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 53
	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.1 - Аналіз використання ресурсів системи

Ресурс	Нормальне навантаження	Пікове навантаження	Рекомендації
Процесор (ЦПУ)	35%	85%	Розглянути оптимізацію коду або апгрейд апаратури.
Оперативна пам'ять	50%	75%	Оптимізувати управління пам'яттю або розширити обсяг.
Дисковий простір	25%	55%	Впровадити стратегії архівації та ротації логів.
Мережеві ресурси	40%	70%	Переглянути мережеву архітектуру та оптимізувати передачу даних.

Ця таблиця демонструє як система використовує апаратні ресурси при різних умовах навантаження. Особлива увага звертається на моменти пікових навантажень, коли виявлено потенційні точки перевантаження, що можуть вплинути на стабільність та швидкість реакції системи.

Після проведення аналізу ресурсів, важливо сконцентрувати увагу на тестуванні основних функцій системи, щоб забезпечити їх надійність та відповідність очікуванням користувачів. Цей етап тестування дозволяє виявити і вирішити специфічні проблеми функціоналу, які можуть вплинути на загальну продуктивність та користувацький досвід.

Таблиця 3.2 - Результати тестування основних функцій

Функція	Тестовий сценарій	Результат та Статус
Перегляд полісів	Відкриття сторінки з полісами	Відображення відповідає дизайну - Пройдено
Детальна інформація про поліс	Клік по конкретному полісу	Деталі полісу доступні - Пройдено
Відкриття новин	Перехід в розділ новин	Новини актуалізовані - Пройдено
Огляд послуг	Перегляд різних страхових послуг	Послуги відображено правильно - Пройдено
Відгуки користувачів	Перегляд розділу відгуків	Відгуки доступні та актуальні - Пройдено
Форма замовлення	Заповнення форми замовлення послуг	Форма працює належним чином - Пройдено

Ця таблиця забезпечує компактний огляд тестування основних доступних функцій вашої вебсистеми та їх статусів. Вона дозволяє швидко оцінити, чи функціонал системи працює згідно з вимогами і чи готова система до використання кінцевими користувачами.

Підсумковий аналіз показує, як система справляється з задачами в реальних умовах і виявляє потенційні вузькі місця в архітектурі та коді. На основі отриманих даних складається звіт, який включає опис виявлених проблем, їх вплив на загальну роботу системи та рекомендації щодо їх усунення. Завершується процес тестування формуванням рекомендацій щодо оптимізації і покращення програмного продукту, що дозволяє підвищити загальну надійність і продуктивність системи.

3.5 Рекомендації щодо подальшого розвитку системи

На підставі проведеного аналізу та тестування веборієнтованої інформаційної системи страхового агентства, можна сформулювати ряд рекомендацій, які допоможуть поліпшити її роботу та ефективність використання.

Технічні аспекти:

- враховуючи високе навантаження на процесор та оперативну пам'ять під час пікових періодів, рекомендується оптимізувати серверні скрипти та використовувати кешування де це можливо;
- розглянути впровадження балансувальників навантаження для рівномірного розподілу трафіку між серверами;
- для зменшення часу завантаження статичних ресурсів на клієнтській стороні можна використати мережу доставки контенту CDN;
- налаштування регулярного бекапу даних для запобігання втрати інформації;

Функціональні вдосконалення:

- підвищити адаптивність інтерфейсу для комфортного користування на мобільних пристроях;
- забезпечити підтримку декількох мов для звернення до більш широкого аудиторії;
- впровадити розширений пошук по сайту для спрощення навігації користувачів;

Маркетингові та SEO заходи:

- проаналізувати та оптимізувати контент для кращого ранжування у пошукових системах;
- регулярно оновлювати блог та новинні сторінки для залучення нових відвідувачів;
- використовувати цільову онлайн-рекламу для просування послуг страхування;

Користувацький досвід (UX):

- налагодити збір зворотного зв'язку для виявлення потенційних поліпшень;
- регулярно проводити А/В тестування для підвищення ефективності елементів інтерфейсу;

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 56
Арк.	№ докум.	Підпис	Дата			

Імплементуючи ці рекомендації, можливо досягти значного підвищення якості сервісу, підвищення задоволеності користувачів, та покращення загального іміджу страхового агентства в цифровому просторі.

3.6 Висновки

У третьому розділі дипломної роботи детально розкривається процес практичної реалізації інформаційної системи для страхової агенції, яка побудована на основі передових технологій програмування — JavaScript і бібліотеки React. Аналіз вибору технологій та архітектури системи ілюструє перетворення теоретичних знань у функціональний та надійний вебзастосунок.

Важливим аспектом розробки є продумана структура бази даних та ефективне використання патерну Модель-Вид-Контролер, який забезпечує чітку сепарацію та взаємодію між сервером та клієнтською частиною. Модульність системи грає ключову роль, надаючи змогу гнучко масштабувати та оптимізувати її відповідно до динамічних вимог ринку та потреб бізнесу.

Значну увагу приділено забезпеченню високої інтегрованості та інтерактивності компонентів системи, що сприяє покращенню взаємодії з користувачами та підвищенню ефективності страхових послуг. У результаті, створена інформаційна система відзначається стабільністю, високим рівнем безпеки та якісного обслуговування користувачів.

Розроблена в рамках дипломної роботи система не тільки відповідає сучасним технічним стандартам вебзастосунків, але й має широкі перспективи для подальшої модернізації в контексті постійної еволюції страхового бізнесу та прогресу в області інформаційних технологій. Завдяки гнучкій архітектурі, система може бути адаптована до нових вимог, забезпечуючи тривалу актуальність та конкурентоздатність на ринку страхових послуг.

ВИСНОВКИ

У дипломній роботі, на основі виконаних теоретичних досліджень та практичної розробки, було досягнуто ключових цілей та вирішено поставлені завдання щодо створення веборієнтованої інформаційної системи для страхового агентства. Розробка системи базувалася на сучасних технологіях та методологіях програмування, зокрема, використанні мови програмування Js, фреймворку React та СУБД NoSql Web Api, що забезпечило високий рівень безпеки, надійності та зручності користувачів.

У першому розділі було проведено детальний аналіз існуючих інформаційних систем страхових компаній та визначено основні вимоги та функціонал, необхідний для ефективного функціонування системи страхового агентства. Також було обґрунтовано вибір технологічного стеку для реалізації проєкту.

У другому розділі здійснено проектування архітектури системи, описано структуру бази даних та взаємозв'язки між її таблицями. Було розроблено діаграми взаємодії компонентів системи та визначено основні модулі та їх функції, що дозволило забезпечити гнучкість та масштабованість проєкту.

У третьому розділі викладено процес реалізації системи, включаючи детальний опис розробки модулів системи та їх взаємодії. Було продемонстровано практичне застосування архітектурного патерну Модель-Вид-Шаблон, що забезпечило чітку організацію коду та спрощення подальшої підтримки системи. Також було виконано тестування системи, що підтвердило її ефективність та готовність до впровадження.

Загалом, розроблена інформаційна система страхового агентства відповідає сучасним вимогам до вебзастосунків, забезпечує високу швидкість обробки даних, безпеку зберігання інформації та зручність використання користувачами. Реалізація проєкту сприятиме підвищенню ефективності роботи страхового агентства, оптимізації бізнес-процесів та покращенню якості обслуговування

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						58
Арк.	№ докум.	Підпис	Дата			

клієнтів. Практична частина демонструє здатність системи до масштабування та адаптації під зростаючі потреби користувачів і бізнесу.

Завершення роботи над цим проектом відкриває шлях для подальших досліджень і розвитку в області інформаційних систем для страхових агентств, зокрема, інтеграції з іншими фінансовими системами, впровадження штучного інтелекту для автоматизації процесів обробки даних та підвищення ефективності взаємодії з клієнтами.

					КвРІСТ 190119.21.01.03 ПЗ	Арк.
						59
	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Bach J. Programação javascript: javascript para iniciantes. Independently Published, 2021.
2. Collins M. J. Cascading style sheets. *Pro HTML5 with CSS, JavaScript, and Multimedia*. Berkeley, CA, 2017. P. 15–27. URL: https://doi.org/10.1007/978-1-4842-2463-2_2 (date of access: 20.06.2024).
3. Collins M. J. Cascading style sheets. *Pro HTML5 with visual studio 2015*. Berkeley, CA, 2015. P. 79–118. URL: https://doi.org/10.1007/978-1-4842-1147-2_4 (date of access: 20.06.2024).
4. Daggett M. E. Expert JavaScript. Berkeley, CA : Apress, 2013. URL: <https://doi.org/10.1007/978-1-4302-6098-1> (date of access: 20.06.2024).
5. Escribano L. F.-P. JavaScript. lulu.com, 2015. 78 p.
6. Ferguson R. Beginning JavaScript. Berkeley, CA : Apress, 2019. URL: <https://doi.org/10.1007/978-1-4842-4395-4> (date of access: 20.06.2024).
7. Ferguson R., Cirkel K. JavaScript recipes. Berkeley, CA : Apress, 2017. URL: <https://doi.org/10.1007/978-1-4302-6107-0> (date of access: 20.06.2024).
8. Gandhi R. JavaScript next. Berkeley, CA : Apress, 2019. URL: <https://doi.org/10.1007/978-1-4842-5394-6> (date of access: 20.06.2024).
9. Gilorien. DHTML and javascript: cascading style sheets, javascript style sheets, & javascripted layers plus advanced javascript 1.2 & 1.3. Upper Saddle River, NJ : Prentice Hall PTR, 2000. 1142 p.
10. Gosselin D. Javascript. 5th ed. Boston, Mass : Course Technology, 2011. 899 p.
11. Hudson S. JavaScript creativity. Berkeley, CA : Apress, 2014. URL: <https://doi.org/10.1007/978-1-4302-5945-9> (date of access: 20.06.2024).
12. Mason P. JavaScript. *SAS stored processes*. Berkeley, CA, 2020. P. 65–80. URL: https://doi.org/10.1007/978-1-4842-5925-2_4 (date of access: 20.06.2024).
13. Mayer G. E. T., Awesomeness J. JavaScript: javascript awesomeness book. CreateSpace Independent Publishing Platform, 2017. 68 p.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 60
Арк.	№ докум.	Підпис	Дата			

14. Negrino T. JavaScript. 8th ed. Berkeley, CA : Peachpit Press, 2012. 518 p.
15. Pomaska G. JavaScript. *Webseiten-Programmierung*. Wiesbaden, 2012. P. 79–126. URL: https://doi.org/10.1007/978-3-8348-2485-1_3 (date of access: 20.06.2024).
16. Pouncey I. Beginning CSS: cascading style sheets for web design. 3rd ed. Indianapolis, Ind : Wiley Pub., Inc., 2011. 432 p.
17. Steyer R. JavaScript. München : Carl Hanser Verlag GmbH & Co. KG, 2014. URL: <https://doi.org/10.3139/9783446439474> (date of access: 20.06.2024).
18. Vickler A. Javascript: javascript basics for beginners. Independently Published, 2021.
19. Vickler A. Javascript: javascript front end programming. Independently Published, 2021.
20. Vodnik S. JavaScript. Cengage Learning, 2014. 896 p.
21. Pomaska G. JavaScript. *Webseiten-Programmierung*. Wiesbaden, 2012. P. 79–126. URL: https://doi.org/10.1007/978-3-8348-2485-1_3 (date of access: 20.06.2024).
22. Ramtal D., Dobre A. JavaScript and HTML5 canvas basics. *Physics for javascript games, animation, and simulations*. Berkeley, CA, 2014. P. 11–35. URL: https://doi.org/10.1007/978-1-4302-6338-8_2 (date of access: 20.06.2024).
23. Steyer R. JavaScript-Techniken bei HTML5 und mehr. *JavaScript*. München, 2014. P. 383–452. URL: <https://doi.org/10.3139/9783446439474.014> (date of access: 20.06.2024).
24. Гладчук О. М. Небанківські фінансові інститути на фінансовому ринку України : автореф. дис. ... канд. екон. наук : 08.00.08. Київ, 2010. 20 с.
25. Залєтов О. М. Державна політика на страховому ринку України. *Фінанси України*. 2001. № 11. С. 119–126.
26. Гончаренко О. Класифікація небанківських фінансових установ як інструмент реалізації стратегії розвитку фінансового сектору економіки

					КВРІСТ 190119.21.01.03 ПЗ	Арк. 61
	Арк.	№ докум.	Підпис	Дата		

України. *Збірник наукових праць Черкаського державного технологічного університету. Серія "Економічні науки"*. 2017. Вип. 44, ч. 1. С. 15–25.

27. Калугарь А., Кучерівська С. Особливості взаємодії учасників страхування в період війни. *Економіка та суспільство*. 2022. № 46. URL: <https://doi.org/10.32782/2524-0072/2022-46-15> (дата звернення: 22.06.2024).

28. Клапків Ю. М. Функції ринку страхових послуг: теоретико-методологічна концептуалізація. *Збірник наукових праць донецького державного університету управління. серія "економіка"*. 2017. Т. 18, вип. 301. С. 60–66.

29. Кучерівська С. С. Вплив інформаційних технологій на розвиток страхових послуг. *Проблеми сучасних трансформацій. Серія: економіка та управління*. 2024. № 11. URL: <https://doi.org/10.54929/2786-5738-2024-11-08-03> (дата звернення: 22.06.2024).

30. Мельничук Ю. Аналіз факторів впливу на ринок страхових послуг зі страхування життя. *Бухгалтерський облік і аудит*. 2018. № 2. С. 18–24.

31. Прокопович-Павлюк І., Федюра І. Оцінка діяльності страхових компаній України. *Ефективне управління економікою на світовому, державному та регіональному рівнях*. 2024. URL: <https://doi.org/10.36059/978-966-397-367-8-30> (дата звернення: 22.06.2024).

32. Іонін Є., Беспалова А. Показники надійності страхових компаній України. *Débats scientifiques et orientations prospectives du développement scientifique*. 2021. URL: <https://doi.org/10.36074/logos-05.02.2021.v1.28> (дата звернення: 22.06.2024).

33. Законодавство України про страхування. Київ : КНТ, вид-во, 2003. 260 с.

34. Про внесення зміни до статті 11 Закону України "Про загальнообов'язкове державне пенсійне страхування": Закон України від 10.01.2006 р. № 3297-IV. URL: <https://zakon.rada.gov.ua/laws/show/3297-15#Text> (дата звернення: 22.06.2024).

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 62
Арк.	№ докум.	Підпис	Дата			

35. Баглюк Ю. Б. Страховий ринок України: капіталізація й конкуренція. *Фінанси України*. 2012. № 4 (197). С. 67–75.
36. Любкіна О. В. Фінансові ринки у контексті глобалізації. *Фінанси України*. 2005. № 9. С. 122–128.
37. Марина А. С., Пеценко М. В. Страховий ринок України в умовах війни. *Цифрова економіка та економічна безпека*. 2023. № 5 (05). С. 44–51. URL: <https://doi.org/10.32782/dees.5-7> (дата звернення: 22.06.2024).
38. Born P., Vujakowski D. Economic transition and insurance market development: evidence from post-communist European countries. *The geneva risk and insurance review*. 2021. URL: <https://doi.org/10.1057/s10713-021-00066-3> (date of access: 22.06.2024).
39. Born P., Vujakowski D. Insurance research in Central, Eastern, and Southeastern Europe: What we can learn from XPRIMM data. *Risk management and insurance review*. 2022. URL: <https://doi.org/10.1111/rmir.12214> (date of access: 22.06.2024).
40. Шкарлет С. М. Інформаційна економіка як основа соціально-економічної регенерації України. *Економіка України*. 2014. № 11 (636). С. 51–60.
41. Клименко О. В. Напрями вдосконалення регулювання ринку страхування. *Економіка України*. 2019. № 7/8 (692/693), лип. - серп. С. 55–70.
42. Лопатіна О. В. Сучасний стан та перспективи розвитку страхового ринку в Україні : thesis. 2018. URL: <http://dspace.oneu.edu.ua/jspui/handle/123456789/9752> (дата звернення: 22.06.2024).
43. Чайка В. В., Чуйкова О. О. Проблемні аспекти пенсійного страхування: національний та міжнародний вимір. *Гуманітарний і інноваційний ракурс професійної майстерності: пошуки молодих вчених*. 2022. С. 205–207. URL: <https://doi.org/10.36059/978-966-397-266-4/57> (дата звернення: 22.06.2024).
44. Бунін С. В. Розвиток світового ринку страхових послуг в умовах глобальної конкуренції : thesis. 2020.

					КвРІСТ 190119.21.01.03 ПЗ	Арк. 63
Арк.	№ докум.	Підпис	Дата			

URL: <http://dspace.univer.kharkov.ua/handle/123456789/15316> (дата звернення: 22.06.2024).

45. Гаманкова О. О. Облік і аудит у страхових організаціях : Навч.-метод. посіб. для самот. вивч. дисципліни. Київ : КНЕУ, 2004. 183 с.

46. Гаманкова О. О. Страхування як вид підприємницької діяльності на ринку фінансових послуг: зміст та необхідність законодавчого визначення. *Вісник київського національного університету імені тараса шевченка. економіка*. 2005. Вип. 80. С. 48–51.

47. Страховий ринок України : thesis / Ю. В. Горобець та ін. 2007. URL: <http://essuir.sumdu.edu.ua/handle/123456789/18949> (дата звернення: 22.06.2024).

48. Журавка О. С. Сучасні тенденції розвитку страхового ринку України. *Вісник Української академії банківської справи*. 2010. № 1 (28). С. 110–114.

49. Залетов О. М. Страхування : навч. посіб. Київ : Міжнар. агенція "BeeZone", 2003. 320 с.

50. Ільчук П. Г., Коць О. О., Скоропад І. С. Доходи та витрати страховиків: динаміка, чинники впливу та особливості прогнозування. *Efektivna ekonomika*. 2022. № 8. URL: <https://doi.org/10.32702/2307-2105.2022.8.10> (дата звернення: 22.06.2024).

51. Козьменко О. В. Страховий ринок України у контексті сталого розвитку : монографія. Суми : ДВНЗ "УАБС НБУ", 2008. 350 с.

52. Шулешова І. В. Розвиток страхового ринку України. *Вісник Сумського національного аграрного університету. Серія: фінанси і кредит*. 2012. № 2 (33). С. 240–246.

53. Ліхоносова Г. С., Міняйло В. О. Управління процесом оподаткування страхової діяльності. *Agrosvit*. 2023. № 23. С. 28–33. URL: <https://doi.org/10.32702/2306-6792.2023.23.28> (дата звернення: 22.06.2024).

					КВРІСТ 190119.21.01.03 ПЗ	Арк. 64
Арк.	№ докум.	Підпис	Дата			

54. Навроцький С. А. Страхування як самостійна економічна категорія відтворення. *Економіка АПК*. 2006. № 9. С. 37–44.

55. Ткаченко Н. В. Розвиток інформаційних технологій як один із вирішальних чинників підтримання стійкості страхових компаній : thesis. 2008. URL: <http://essuir.sumdu.edu.ua/handle/123456789/61716> (дата звернення: 22.06.2024).

56. Снідевич О. С. Щодо доцільності прийняття проекту Закону України "Про внесення змін до деяких законодавчих актів України (щодо виконання судових рішень)" за реєстр. № 10142 від 1 березня 2012 року. *Юриспруденція: теорія і практика*. 2011. № 6 (80). С. 63–64.

57. Крихівська Н. О., Ромашко О. М., Мацьків Р. Т. Ефективність діяльності страхових компаній на ринку медичного страхування. *Проблеми сучасних трансформацій. Серія: економіка та управління*. 2023. № 9. URL: <https://doi.org/10.54929/2786-5738-2023-9-08-01> (дата звернення: 22.06.2024).

58. Школьник І. О., Школьник І. А., Shkolnyk I. O. Стратегія розвитку фінансового ринку України : thesis. 2008. URL: <http://essuir.sumdu.edu.ua/handle/123456789/51680> (дата звернення: 22.06.2024).

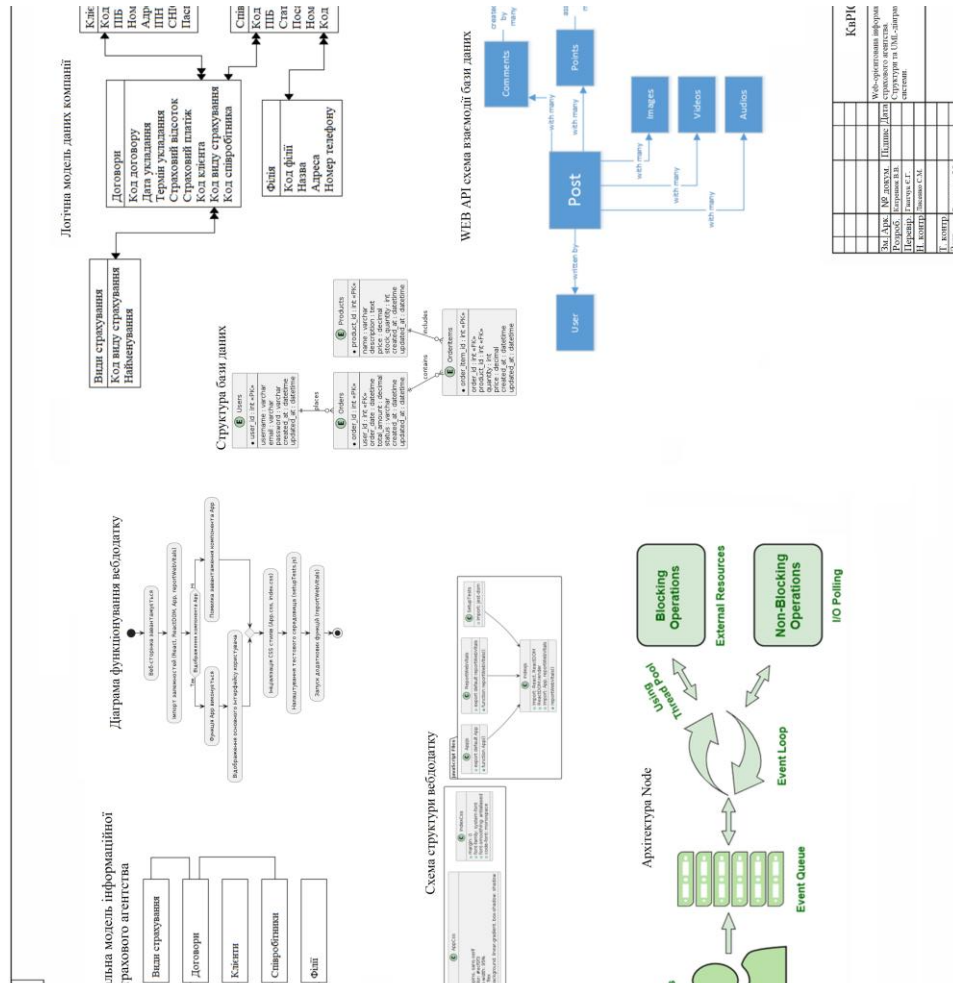
59. Стешенко О. Д., Мінаєва О. О. Страхування в системі управління банківськими ризиками. *Вісник економіки транспорту і промисловості*. 2016. № 54. URL: <https://doi.org/10.18664/338.47:338.45.v0i54.76463> (дата звернення: 22.06.2024).

60. Riashchenko V., Kremen V., Bochkarova T. A discriminant analysis of insurance companies in ukraine. *Financial markets, institutions and risks*. 2017. Vol. 1, no. 4. P. 65–73. URL: [https://doi.org/10.21272/fmir.1\(4\).65-73.2017](https://doi.org/10.21272/fmir.1(4).65-73.2017) (date of access: 22.06.2024).

					КВРІСТ 190119.21.01.03 ПЗ	Арк. 65
	Арк.	№ докум.	Підпис	Дата		

Додаток А (обов'язковий)

Копія креслення «Структури та UML-діаграми інформаційної системи»



Код	Назва	Тип	Категорія
1	Договори	Таблиця	Основні дані
2	Види страхування	Таблиця	Основні дані
3	Квитки	Таблиця	Основні дані
4	Співробітники	Таблиця	Основні дані
5	Філії	Таблиця	Основні дані
6	Пользователи	Таблиця	Основні дані
7	Посты	Таблиця	Основні дані
8	Комментарии	Таблиця	Основні дані
9	Изображения	Таблиця	Основні дані
10	Видео	Таблиця	Основні дані
11	Аудио	Таблиця	Основні дані

Додаток Б (обов'язковий)

Копія креслення «Підсумковий аналіз тестувань та рекомендацій»

Аналіз використання ресурсів системи

Ресурс	Нормальне навантаження	Пікове навантаження	Рекомендації
Процесор (ЦПУ)	35%	85%	Розглянути оптимізацію коду або апгрейд апаратури.
Оперативна пам'ять	50%	75%	Оптимізувати управління пам'яттю або розширити обсяг.
Дисковий простір	25%	55%	Впровадити стратегії архівації та ротачії логів.
Мережеві ресурси	40%	70%	Переглянути мережеву архітектуру та оптимізувати передачу даних.

Результати тестування основних функцій

Функція	Тестовий сценарій	Результат та С
Перегляд посвід	Відкриття сторінки з посвідом	Відображення посвіда Пройдено
Детальна інформація про посід	Клік по конкретному посіву	Детальна посвід Пройдено
Відкриття новин	Перехід в розділ новини	Новини актуалізовані Пройдено
Огляд посвід	Перегляд різних страхових посвід	Посвідта відображено Пройдено
Відгуки користувачів	Перегляд розділу відгуків	Відгуки доступні та актуальні Пройдено
Форми заповнення	Заповнення форми заповнення посвід	Форми працює належно Пройдено

Картка			
№	Ім'я	Піде	Дата
№1	№1	№1	№1
№2	№2	№2	№2
№3	№3	№3	№3
№4	№4	№4	№4
№5	№5	№5	№5
№6	№6	№6	№6
№7	№7	№7	№7
№8	№8	№8	№8
№9	№9	№9	№9
№10	№10	№10	№10
№11	№11	№11	№11
№12	№12	№12	№12
№13	№13	№13	№13
№14	№14	№14	№14
№15	№15	№15	№15
№16	№16	№16	№16
№17	№17	№17	№17
№18	№18	№18	№18
№19	№19	№19	№19
№20	№20	№20	№20
№21	№21	№21	№21
№22	№22	№22	№22
№23	№23	№23	№23
№24	№24	№24	№24
№25	№25	№25	№25
№26	№26	№26	№26
№27	№27	№27	№27
№28	№28	№28	№28
№29	№29	№29	№29
№30	№30	№30	№30
№31	№31	№31	№31
№32	№32	№32	№32
№33	№33	№33	№33
№34	№34	№34	№34
№35	№35	№35	№35
№36	№36	№36	№36
№37	№37	№37	№37
№38	№38	№38	№38
№39	№39	№39	№39
№40	№40	№40	№40
№41	№41	№41	№41
№42	№42	№42	№42
№43	№43	№43	№43
№44	№44	№44	№44
№45	№45	№45	№45
№46	№46	№46	№46
№47	№47	№47	№47
№48	№48	№48	№48
№49	№49	№49	№49
№50	№50	№50	№50
№51	№51	№51	№51
№52	№52	№52	№52
№53	№53	№53	№53
№54	№54	№54	№54
№55	№55	№55	№55
№56	№56	№56	№56
№57	№57	№57	№57
№58	№58	№58	№58
№59	№59	№59	№59
№60	№60	№60	№60
№61	№61	№61	№61
№62	№62	№62	№62
№63	№63	№63	№63
№64	№64	№64	№64
№65	№65	№65	№65
№66	№66	№66	№66
№67	№67	№67	№67
№68	№68	№68	№68
№69	№69	№69	№69
№70	№70	№70	№70
№71	№71	№71	№71
№72	№72	№72	№72
№73	№73	№73	№73
№74	№74	№74	№74
№75	№75	№75	№75
№76	№76	№76	№76
№77	№77	№77	№77
№78	№78	№78	№78
№79	№79	№79	№79
№80	№80	№80	№80
№81	№81	№81	№81
№82	№82	№82	№82
№83	№83	№83	№83
№84	№84	№84	№84
№85	№85	№85	№85
№86	№86	№86	№86
№87	№87	№87	№87
№88	№88	№88	№88
№89	№89	№89	№89
№90	№90	№90	№90
№91	№91	№91	№91
№92	№92	№92	№92
№93	№93	№93	№93
№94	№94	№94	№94
№95	№95	№95	№95
№96	№96	№96	№96
№97	№97	№97	№97
№98	№98	№98	№98
№99	№99	№99	№99
№100	№100	№100	№100

Ім'я користувача:
Кафедра КІ

Дата перевірки:
10.06.2024 17:45:27 EEST

Дата звіту:
10.06.2024 17:54:44 EEST

ID перевірки:
1016343473

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Катренко_Web-орієнтована інформаційна система страхового агентства

Кількість сторінок: 72 Кількість слів: 11369 Кількість символів: 93416 Розмір файлу: 9.39 MB ID файлу: 1016144925

2.95% Схожість

Найбільша схожість: 0.54% з джерелом з Бібліотеки (ID файлу: 1016107776)



0.03% Цитат



0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.



Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 9%

ID: 129497 Назва: БКР Web-орієнтована інформаційна система страхового агентства Додано в БД: 2024-06-10 Автора: В.В. Катренко Керівники: Є.Г. Гнатчук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	80973	628	3418 (4%)	47 (7%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Катренюк Владислав Валерійович

Тема: Web-орієнтована інформаційна система страхового агентства

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3; кількість сторінок записки 58

1. Короткий зміст роботи та прийнятих рішень: Розроблена web-орієнтована інформаційна система страхового агентства, яка покращує процеси управління ризиками, ефективність страхових стратегій та якість обслуговування клієнтів.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було проаналізовано предметну область. У першому розділі розглянуто створення web-орієнтованої інформаційної системи страхового агентства. Визначено ключові завдання для її ефективного функціонування, включаючи аналіз потреб та викликів. Обрано оптимальний набір технологій для вирішення поставлених завдань.

В другому розділі було проведено проектування та розроблена архітектура інформаційної системи. Розглянуті функційні, нефункційні, а також користувацькі вимоги.

У третьому розділі дипломної роботи детально розкривається процес практичної реалізації інформаційної системи для страхової агенції, яка побудована на основі сучасних технологій програмування. Аналіз вибору технологій та архітектури системи ілюструє перетворення теоретичних знань у функціональний та надійний вебдодаток.

4. Позитивні сторони роботи: Значну увагу приділено забезпеченню високої інтегрованості та інтерактивності компонентів системи, що сприяє покращенню взаємодії з користувачами та підвищенню ефективності страхових послуг. У результаті, створена інформаційна система відзначається стабільністю, високим рівнем безпеки та якісного обслуговування користувачів.

5. Негативні сторони роботи: В роботі не приділено достатньої уваги аналізу переваг та недоліків існуючих систем.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому
професійному рівні

8. Інші зауваження: =

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота
заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Резуш Микола Васильович, доцент кафедри АкіТорР, ВНУ

"19" 06 2024р.



Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорушенко Т. О.

Катренюка Владислава Валерійовича
ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-20-1

ЗАЯВА

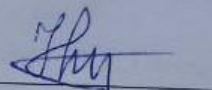
З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

20.06.24

дата



підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Web-орієнтована інформаційна система страхового агентства
 Автор: Катренко Владислав Валерійович
 Спеціальність: 126 – Інформаційні системи та технології
 Освітня програма: освітньо-професійна
 Науковий керівник: Гнатчук Є.Г., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення, розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості Unicheck, складає 2.95% і адресується до 106 першоджерела; та системою Anti-Plagiarism складає 1%.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

Є. Г. Гнатчук

Є. Г. Гнатчук

Т. О. Говорущенко