




КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Веб орієнтовна система обміну текстовими повідомленнями

Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань
Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності
Освітня програма Комп'ютерні науки
Назва освітньої програми

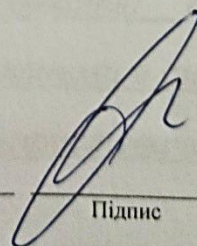
Виконав: студент 4 курсу, група КН-17-2  В.А. Козакевич
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ  О.В. Мазурець
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КНІТ, д.т.н., професор

08 червня 2021 р.


Підпис

О.В. Бармак
Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра комп'ютерних наук та інформаційних технологій

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук та інформаційних технологій

(підпис)

д.т.н., професор О.В. Бармак

«08» лютого 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Веб орієнтовна система обміну текстовими повідомленнями»

2. Завдання видано студенту Козакевичу Віталію Анатолійовичу
(прізвище, ім'я, по батькові)

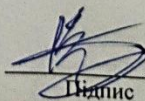
3. Керівник роботи доцент кафедри КНІТ Мазурець Олександр Вікторович
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «05» 02 2021 р. № 11

5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

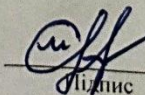
Мета роботи – розробка веб орієнтовної системи обміну текстовими повідомленнями, яка виконує функції реєстрації і авторизації користувачів, роботи з профілем користувача, роботи з відправленням повідомлень, пошук і перегляд відкритої інформації профілів інших користувачів тощо. Має бути одержаний програмний продукт із можливістю обміну текстовими повідомленнями між користувачами, зареєстрованими в системі, з додатковою можливістю збереження всіх переписок між користувачами в базі даних для подальшого використання.

Виконавець: студент 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

В.А. Козакевич
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КНІТ
Науковий ступінь, посада


Підпис

О.В. Мазурець
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Веб орієнтовна система обміну текстовими повідомленнями»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-17-2 Козакевич Віталій Анатолійович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КНІТ Мазурець Олександр Вікторович

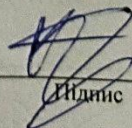
Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
57	23	7	29	2

Метою кваліфікаційної роботи бакалавра є розробка на платформі PHP веб орієнтовної системи для обміну текстовими повідомленнями текстовими повідомленнями між користувачами, зареєстрованими в системі. Веб орієнтовану систему обміну текстовими повідомленнями було створено на мові програмування PHP з використанням програмного каркасу Laravel СКБД MySQL, редактор програмного коду Visual Studio Code.

Одержана в результаті веб орієнтовна система обміну текстовими повідомленнями виконує функції: реєстрація користувачів та робота з ними (реєстрація і авторизація користувачів); робота з особистими даними користувачів (редагування даних профілю); робота з повідомленнями (відправлення, одержання, формування діалогів, переліку діалогів); робота з профілями інших користувачів (пошук, перегляд відкритої інформації).

Виконавець: студент 4 курсу, група КН-17-2
Курс, група виконавця


Підпис

В.А. Козакевич
Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1	
Характеристика предметної області та постановка задачі	7
1.1 Аналіз предметної області	7
1.2 Аналіз існуючого програмного забезпечення предметної області	11
1.2.1 Інформаційні системи обміну повідомленнями	11
1.2.1 Можливості та недоліки існуючих систем обміну повідомленнями	15
1.3 Аналіз сучасних засобів створення програмного забезпечення	18
1.4 Постановка задачі та вимоги до розробки інформаційної системи.....	20
Розділ 2	
Проектування інформаційної системи	21
2.1 Функціональна структура та бізнес-процеси системи	21
2.2 Інформаційна структура системи	24
2.3 Вибір засобів розробки інформаційної системи	27
2.3.1 Вибір мови програмування	27
2.3.2 Вибір фреймворку	30
2.3.3 Вибір редактора програмного коду.....	32
2.3.4 Вибір СКБД	33
Розділ 3	
Програмна реалізація інформаційної системи	35
3.1 Структура та функціональне призначення складових системи	35
3.2 Особливості реалізації складових системи	36
3.3 Тестування інформаційної системи	45
3.4 Інструкція користувача.....	48
3.5 Вимоги до розгортання інформаційної системи.....	54
Висновки	55
Перелік посилань.....	56
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
ІС	Інформаційна система
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
КНІТ	Комп'ютерні науки та інформаційні технології
Git	Розподілена система керування версіями файлів
ПП	Програмний продукт
API	Програмний інтерфейс додатку
Wi-Fi	Бездротове з'єднання з мережею інтернет
MVC	Архітектурний шаблон (модель-уявлення-контролер)

Вступ

В сьогоднішній день спілкування відіграє одну із найважливіших ролей в житті людини. Так, потреба в спілкуванні є впродовж всього життя, а з появою вільного доступу до мережі Інтернета, виникла потреба в створенні веб орієнтованого додатку по обміну текстовими повідомленнями, для комунікації між людьми із різних куточків світу.

Невербальні способи спілкування потрібні, в першу чергу, для того, щоб регулювати та стабілізувати всі процеси спілкування, створювати соціальний контакт між людьми, проявити емоції, дати об'єктивну оцінку будь-якій ситуації. Зазвичай, ці методи не можуть самотужки передавати пряме значення слів, за таким винятком як деякі символів або жестів. Невербальні засоби точно скоординовані між собою і словесними текстами. Не точність деяких невербальних засобів знатною мірою ускладнює спілкування між людьми. Невербальні види спілкування, на відміну від живого спілкування, між людьми, які говорять, а також та, яка сприймає інформацію, може усвідомити не точною мірою .

На даний момент існує безліч додатків для комунікації між різними людьми, в різних країнах. Така комунікація може бути як у вигляді текстових повідомлень відправлених від одного користувача іншому, так і аудіо зв'язок, а в деяких випадках і навіть відео зв'язок [1], що допомагає проводити відео конференції з будь якої частини світу. Звісно технології не стоять на місці, а найближчим часом, може появиться новий спосіб комунікації, такий як невербальний.

Найбільше користуються популярність саме додатки з можливістю обміну текстовими повідомленнями між користувачами у форматі переписки. Також між ними є можливість створення чатів і діалогів, що дозволяє користувачам спілкуватися групами. Соціальні мережі популярні завдяки своїм перевагам, які не можна ігнорувати, адже завдяки їм користувачі так сильно вподобали і до сих пір не перестають спілкуватися в Інтернеті.

Одними із найпопулярніших середовищ обміну текстовими повідомленнями являються месенджери. Месенджер – це програма, яка може використовуватися як на телефоні, так і на персональних комп'ютерах, для миттєвого обміну текстовими повідомленнями між користувачами, які були зареєстровані в системі. Зазвичай такі програми безплатні і працюють тільки з доступом до інтернету [2]. Обмін текстовими повідомленнями між користувачами, які попередньо зареєструвалися в системі, відбувається безпосередньо у вигляді діалогу між двома користувачами, із зазначенням конкретного часу відправлення повідомлення. Месенджери надають безпосередньо можливість створення групових чатів з людьми які мають схожі інтереси. Також не мало важна можливість додавання до текстового повідомлення медіа фалів. Таким способом можна ділитися фотографіями або відео із своїми друзями. Система не обмежує користувача у часі, тому її використання може буде необмежене. На даний момент, більшість сучасних месенджерів підтримують функцію обміну аудіо повідомленнями, що на сам перед полегшує роботу користувача з додатком. В деяких окремих випадках є можливість дзвінка з месенджера на стаціонарний телефон, але з додатковою платою.

Відправлення повідомлення через месенджер використовують інтернет-трафік, і за цей трафік користувач повинен заплатити своєму мобільно оператору. Но в сучасних реаліях, такі веб додатки використовують мінімальну кість мобільного трафіку, для обробки конкретно вашої інформації, якщо не йде надсилання медіа файлу.

В таких веб додатках є один вагомий мінус. Для спілкування із своїм другом, у вас повинен бути встановлений однаковий месенджер, так як, на даний момент, відправка повідомлення з одного месенджера на другий, не є можливою.

Також на сьогоднішній день існує велика кількість різновидів веб орієнтованих додатків по обміну інформацією. Всі вони відрізняються в певній мірі функціоналом, но залишаються за собою одну спільну мету, це обмін інформацією між користувачами [3]. Ця інформація може бути в різному

форматі, так як аудіо, відео або текст. Такі додатки широко використовуються у всіх галузях виробництва і не тільки.

Тому в наш час це актуальна тема, яка потребує уваги із сторони розробників в плані модернізації і оптимізації додатків, для максимальної швидкодії, до в наш час є одним із головних признаков хорошего додатку. Тому було прийняте рішення створити і оптимізувати веб орієнтований додаток по обміну текстовими повідомленнями для зручної роботи з користувачами.

Розділ 1

Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Спілкування – це процес передавання й сприймання повідомлень за допомогою вербальних і невербальних засобів, що охоплює обмін інформацією між учасниками спілкування, її сприйняття й пізнання, а також їхній вплив один на одного і взаємодією щодо досягнення змін у діяльності.

У наші дні, коли інформаційні технології досягли найвищої точки свого розвитку і кожного дня стають все більш простими і приємними для використання користувачем [1] спілкування через Інтернет стало не від'ємною частиною кожної сучасної людини. Для підтримки зв'язку з близькими і рідними, контакту з партнером або в повсякденні рутині, стали широко використовуватися засоби обміну інформації.

Засоби обміну інформацією мають різні види. Основні з них це месенджери, форуми, чати і соціальні мережі.

Одним із найпопулярніших сервісів по обміну інформації, є месенджер. В цього додатку, обмін повідомленнями між користувачами відбувається миттєво і проходить у вигляді діалогу між двома користувачами, у реальному часі. Також є можливість обміну текстовими повідомленнями і з більшою кількістю користувачів [2]. Коли в діалозі більше двох користувачів, то він називається чатом.

Чат це об'єднання трьох і більше користувачів, з можливістю обміну текстовими повідомленнями і медіа файлами між собою. Таке спілкування проходить в режимі реального часу. Основною характеристикою чату є комунікація в реальному часі, або близькому до нього, що відрізняє чат від форумів [3].

Форум це форма спілкування в інтернеті. Кожне повідомлення в форумі має свого автора, зміст і тему для обговорення. На форумах створюють теми для

спілкування або дискусій. Ці теми можуть знаходитися на формі необмежену кількість часу, з можливістю її перегляду в майбутньому [4].

Соціальні мережі це сайт або додаток, з можливістю реєстрації користувача в ній. Після проходження авторизації, користувачу надається можливість створення публічної анкети, з можливістю створення списку своїх друзів, з якими вони можуть підтримувати зв'язок. Також у них є можливість ділитися інформацію з іншими користувачами які слідкують один за одним [5].

З появою вільного доступу до Інтернету з будь якої частин світу та розвитком сучасних технологій соціальні мережі набули особливого значення. Вони стали головною частиною людського життя і займають важливе місце серед захоплень тисяч користувачів. На сучасному Інтернет-ринку існує кілька порталів та мереж, де люди можуть скористатися цією можливістю, щоб безкоштовно спілкуватися між собою. Перелік таких товарів та продуктів, що сприяють такому спілкуванню, поповнюється щодня.

Соціальні мережі популярні завдяки своїм перевагам, які не можна ігнорувати, адже завдяки їм користувачі так сильно вподобали і до сих пір не перестають спілкуватися в Інтернеті. По-перше, є можливість пошуку однодумців. Користувач також може організувати власний блог, де може публікувати потрібну йому інформацію. Така сторінка буде цікава людям з такими ж захопленнями, і ви зможете спілкуватися з людьми, які вас знають.

Месенджер – телекомунікаційна служба для обміну текстовими повідомленнями між комп'ютерами або іншими пристроями користувачів які мають доступ до глобальної мережі через комп'ютерні мережі [5].

Раніше у глобальній мережі користувачі спілкувались переважно через соціальні мережі. Зараз є більш розширені функції. Для цього достатньо портативного комп'ютера, мобільного телефону або планшета. За допомогою цих пристроїв ви можете надсилати документи, файли та повідомлення без використання платних SMS та здійснення телефонних дзвінків. Поява інструментів обміну миттєвими повідомленнями робить це можливим і на самперед дуже зручним.

Слово «месенджер» походить від англійського « messenger», що означає кур'єр, що в свою чергу і означає обмін повідомленнями між різними користувачами, в даному випадку «кур'єром» виступає безпосередньо «messenger». Обмін повідомленнями йде миттєво, в режимі реального часу. Після того, як відправник завершує введення або редагування і натискає кнопку «Відправити», повідомлення негайно відправляється іншій стороні. Але одержувач повідомлення повинен залишатися на зв'язку, інакше повідомлення буде чекати, поки він також не запустить додаток і не зверне на це увагу.

Одна з переваг соціальних мереж – можливість спілкуватися з людьми в будь-який час і в будь-якому місці, коли немає можливості зустрітися віч-на-віч. Це друзі зі школи, які переїхали в інший район, або люди, з якими ми разом вчилися, разом служили і тому подібне.

Якщо у користувача безлімітний Wi-Fi , то всі повідомлення, а також аудіо або відеодзвінки за допомогою месенджера будуть безкоштовними , чи не обмеженими за часом і за кількістю [5].

Якщо на смартфоні інтернет є мобільним, обмеженим по трафіку, то краще виключити спілкування з відео-зв'язку, бо може швидко закінчитися весь місячний оплачений ліміт.

Багато хто знає, що SMS і дзвінки по звичайній стільникового, мобільного зв'язку найчастіше є платними. Але відмовитися від мобільного зв'язку та перейти повністю на месенджери навряд чи можливо. Справа в тому, що зазвичай для реєстрації там аккаунта потрібно підтвердити номер телефону за допомогою SMS.

Крім того, соціальні мережі є об'єктами сучасних музичних творів або старовинних витворів мистецтва, давно забутих пісень. в Інтернеті ви можете знайти велику кількість цікавих фільмів та відео, новин та фото. Ви також можете читати гороскоп щодня безкоштовно. Словом, соціальні мережі відкрили двері в інший цікавий світ, який з часом може викликати таке звикання, після якого в майбутньому ми не зможемо обійтися. Хоча спілкування в Інтернеті сильно відрізняється від реального спілкування, воно по-своєму цікаве.

Перш ніж почати роботу в системі по обміну текстовими повідомлення, користувача потрібно пройти реєстрацію. Реєстрація користувача проходить в режимі реального часу, з потребою вводу особистих даних користувача, з метою його авторизації в системі із подальшою можливістю входу в свій опублікований запис в системі.

При наявності додатку на телефон, потрібно буде встановити його на мобільний телефон і повторно пройти авторизацію в системі. Зазвичай такі додатки не займають багато пам'яті на телефоні, но більшість із них мають систему кешування, яка з часом забруднюється непотрібними файлами, і може бути завантажено, що значно погіршить на продуктивності продукту в цілому.

Наступним кроком користувачу доведеться створити акаунт. Акаунт це запис, що містить набір відомостей, які користувач передає будь-якій комп'ютерній системі. Зазвичай для отримання запису користувачу потрібно пройти реєстрацію в системі, для цього йому потрібно ввести всі потрібні дані. Для ідентифікації користувача використовується логін і пароль. Крім них, акаунт також може зберігати інформацію про дату народження, стать і інтереси користувача, все ця сукупність даних і є акаунтом [4].

Після авторизації користувача в додатку, йому буде доступний інтерфейс користувача, в якому йому буде надана можливість пошуку користувачів по системі, з можливістю відправлення повідомлення цьому користувачу. Також в нього є можливість редагування вказаної інформації, якщо та була не коректно вказана при реєстрації користувача в системі обміну текстовими повідомленнями.

Текстові повідомлення можуть редагуватися і формуватися користувачем, відповідно до його потреб. Вся переписка між користувачами ведеться у формі діалогу, з автоматичною фіксацією часу і дати відправки і повідомлення і з можливістю обрати адресанта.

Якщо виникає потреба у більшій кількості користувачів в діалозі, тоді можна використати функцію створення чатів. В таких чатах може знаходитися

необмежена кількість користувачів, з можливістю обміну текстовими повідомленнями між собою, в рамках цього чату.

В повідомленнях крім тексту, також можуть бути добавлені медіа файли, як для обміну між користувачами, так і для поширення цих файлів в мережі, для загального доступу.

Також доступна можливість архівації діалогів користувачів, для подальшого їхнього збереження.

Отже, основні параметри предметної області наведено у таблиці 1.1.

Таблиця 1.1 – Параметри предметної області

Параметр предметної області	Опис
Користувач	Профіль зареєстрованого користувача в системи, що містить дані про його унікальний номер, логін, список доданих користувачів, особиста інформація.
Повідомлення	Форматований текст, можливість відправки медіа файлів, редагування файлів, очищення діалогів, реакція на повідомлення, можливість надання відповіді на конкретне повідомлення, діалог в якому міститься додаткова інформація про користувача, дата і час відправки повідомлення.

1.2 Аналіз існуючого програмного забезпечення предметної області

1.2.1 Інформаційні системи обміну повідомленнями

На даний момент у цій предметній області є ряд варіацій ІС. Так, соціальні мережі використовують різноманітні технології. Щоб почати відразу писати повідомлення своїм знайомим без будь-яких обмежень, досить завантажити обраний месенджер. Для цього потрібно відкрити програму Google Play [11] для Android [12] системи, App Store [13] для Apple , або Microsoft Store

[14] для Windows Phone. Потім в пошуковому рядку потрібно ввести назву месенджера. Після цього знадобиться натиснути на кнопку «Встановити».

Одним із популярних месенджерів на сьогоднішній день є «Телеграм», який зображений на рисунку 1.1 Він був створений спочатку як експериментальний проект і не планувався бути створеним як месенджер. На його платформа хотіли протестети технологію шифрування листування для користувачів зареєстрованих в системі. Розробникам було цікаво як технологія буде справлятися з великим напором людей і як в загально поведе себе система при великому навантаженні. Саме так і з'явилася перша версія цього додатку, яка зараз пройшла безліч релізів і є доступна майже на всіх площадках.

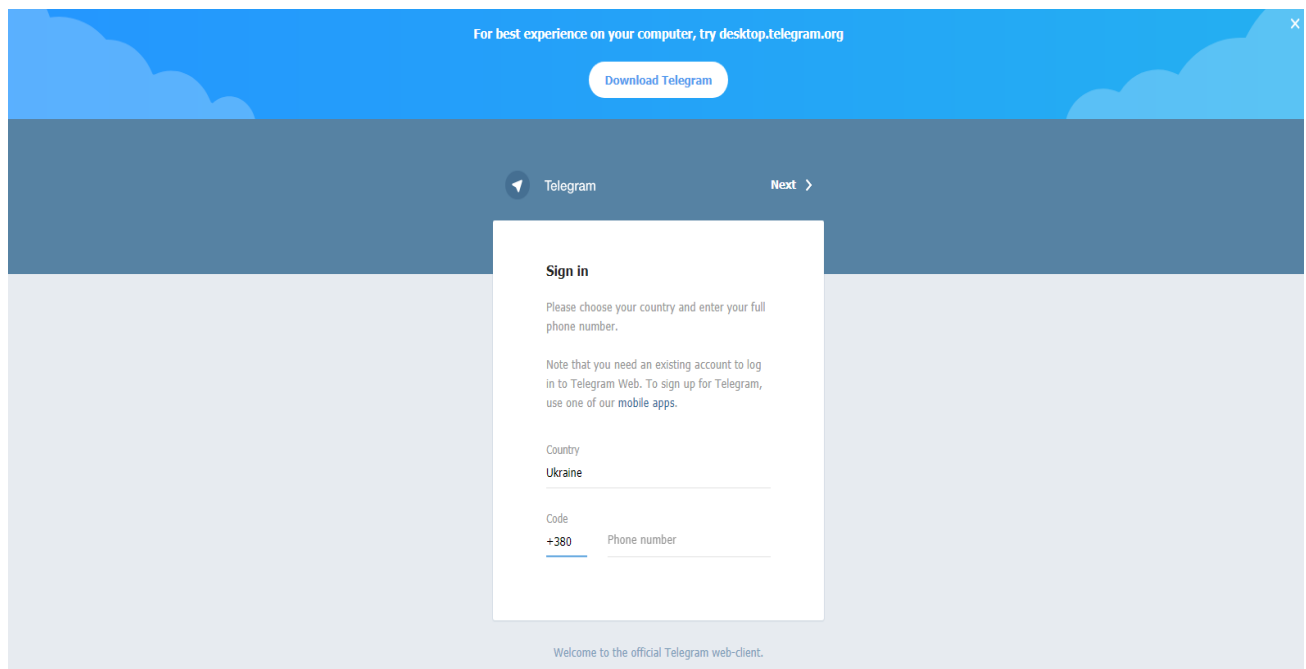


Рисунок 1.1 – Телеграм [16]

Зокрема, на Рисунку 1.2 зображена головна сторінка системи «Whatsapp» [15], яка пропонує можливості завантаження з «Google Play» [11] і «Apple Store» [12] переглядати облікові записи користувачів, ділитися інформацією та оцінювати цю інформацію. Також на цьому сайті є можливість відсилати фото та відео, а також, спілкуватися користувачам між собою.

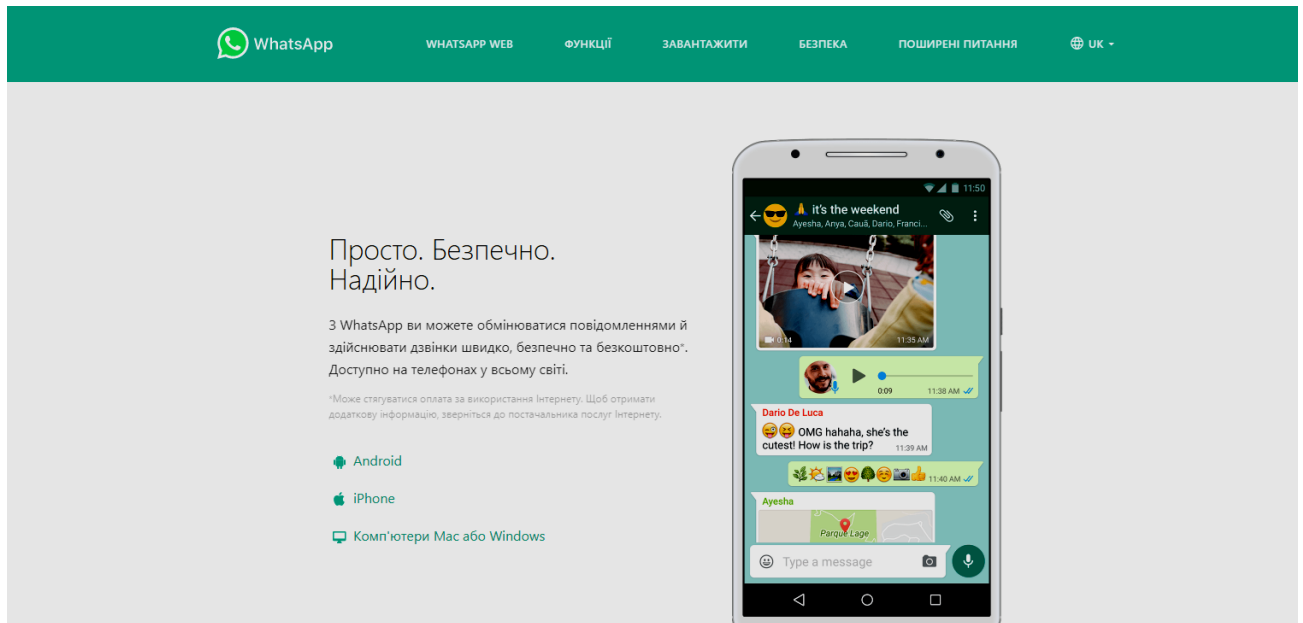


Рисунок 1.2 – Головна сторінка системи «Whatsapp» [15]

На Рисунку 1.3 зображена головна сторінка системи Viber [17]. Система Viber складається з:

- центральної бази даних – ЦБД (користувачі і інформація, яку вони поширюють);
- реєстрація і авторизація користувачів;
- Можливість спілкування між користувачами;
- Можливість оцінювати дописи медіа файли, і ділитися своїми дописами;
- Оцінювати і коментувати повідомлення інших користувачів

Viber був запущений для iPhone в кінці 2010 року, а рання версія додатку для Android була представлена користувачам в наступному році. За функціональністю Viber практично не відрізняється від Skype: голосові і відеодзвінки, а також обмін текстовими повідомленнями мають ті ж функції. Ідентифікація користувача за номером мобільного телефону, додаванню контактів і синхронізації з телефонною книгою з всіма доступними контактами, які коли небудь були добавлені вами, у ваш пристрій.

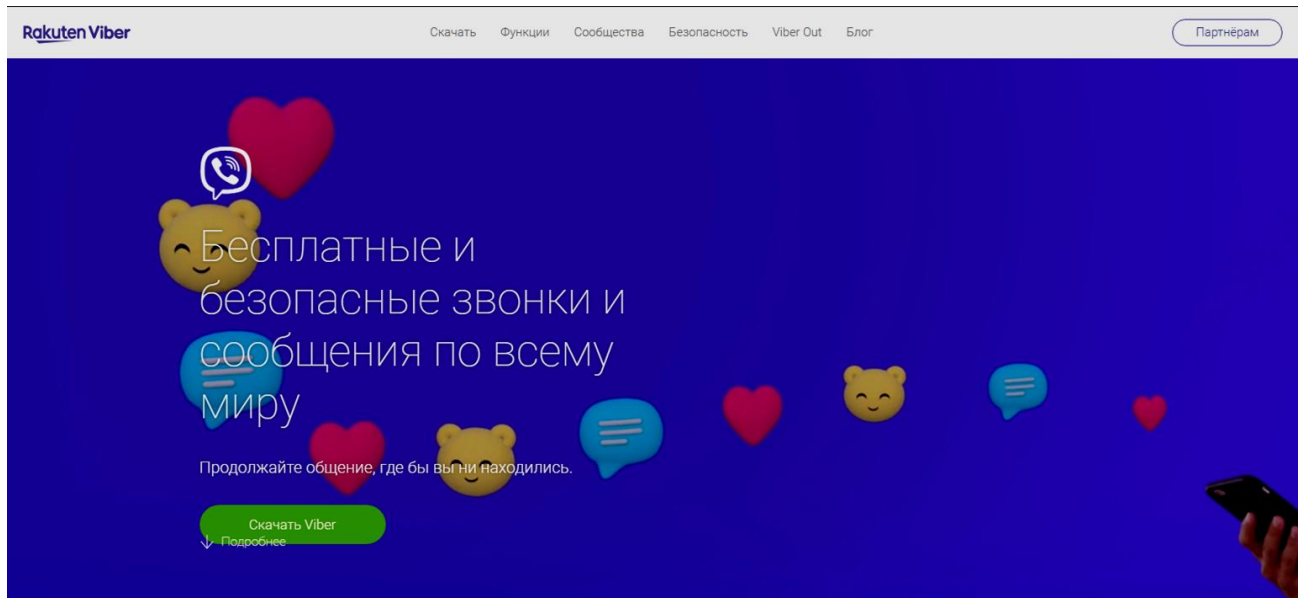


Рисунок 1.3 – Головна сторінка соціальної мережі «Viber» [17]

Порівняно з подібними програмами, Viber використовує менше ресурсів для підтримки умов праці, оскільки вона спеціально розроблена для мобільних пристроїв, що являє собою вагомою перевагою на фоні інших проектів.

Аудиторія Viber [17] налічує більше 200 мільйонів користувачів. Безпека використання програм і зберігання особистої інформації часто піддається різним тестам. У 2013 році навколо даного веб додатку розгорілося кілька скандалів. Навесні було виявлено, що доступ до мобільних пристроїв через веб орієнтований додаток можливий навіть і без захисту паролем. А вже влітку Viber піддався серйозній хакерській атаці, яка вкотре продемонструвала ненадійність сучасного програмного забезпечення.

Ця система:

- здійснює адміністрування та технічну підтримку центральної бази даних (ЦБД);
- забезпечує безперебійну роботу ЦБД;
- забезпечує розробку, оновлення та підтримку програмного забезпечення ЦБД;
- приймає рішення про підключення, відключення та зупинення доступу ЦБД;
- вживає заходів до захисту інформації, що міститься у ЦБД.

Також ця система містить API, яке можна використовувати у своїх програмах.

Тому зараз користувач може знайти безліч прикладів автоматизованих інформаційних соціальних мереж. В основному вони надають користувачам функції спілкування, знаходять правильну інформацію, поширюють корисну інформацію та об'єднують великі групи людей із спільними інтересами. Тому обраний напрям IT є актуальним.

1.2.1 Можливості та недоліки існуючих систем обміну повідомленнями

Аналіз існуючого програмного забезпечення предметної області виявив, що інформаційні системи обміну повідомленнями надають такі основні можливості:

1. Обмін текстовими повідомленнями – найпростіший варіант, на зразок SMS;
2. Можливість прикріплювати смайлики, фотографії, зображення, документи, відео або інші необхідні файли до електронних листів - аналогічно електронних листів, але є тільки онлайн;
3. Створення групи користувачів для спілкування за інтересами, по роботі, по навчанню – колективне спілкування і навіть дистанційне навчання;
4. В деяких месенджерах можна записати голосове повідомлення і відправити його, або є набір голосом - зручно для тих, хто не любить набирати текст вручну;
5. Зустрічається іноді можливість зробити дзвінок з месенджера на мобільний телефон одного або навіть на міський телефон - зазвичай за додаткову плату.
6. Розробники наполегливо працюють над вдосконаленням своєї системи за допомогою нових функцій, таких як наявність хмарного сервера або можливість проведення відео конференцій.

7. Месенджери надають можливість архівації чатів, щоб можна було в разі чого відновити переписку і переговори. Таке може знадобитися, скажімо, при зміні номера телефону абонента.

Із мінусів, можна зазначити:

1. На сьогоднішній день є за надто багато аналогів месенджерів, що не завжди є добрим, так як можливості комунікації між різними месенджерами, на даний момент, не існує.

2. Погана реалізація веб-додатку, значно врізаний функціонал.

3. В деяких випадках, відсутня аудіо і відео зв'язок, між користувачами, які зареєструвалися в додатку.

4. Не завжди вдалий дизайн, на сам перед перенасиченість колярами і надлишковий функціонал.

5. В переважній більшості, дизайнерські рішення і архітектура проекту, запозичені у своїх конкурентів.

6. Не завжди працює синхронізація між девайсами, при прочитаних повідомленнях.

7. Відсутність багато мов локалізації по стандарту.

8. Немає синхронізації з повідомлення і дзвінками девайса користувача, який користується додатком.

9. Відсутня інтеграція з різними пристроями користувачів і синхронізація.

10. Відновлення паролів, реєстрація, і зміна інформації користувача, відбувається з допомогою відправки SMS повідомлення на телефон користувача.

11. Одним із вагомих недоліків, деяких месенджерів, є їхня кросплатформеність.

12. Обмежені можливості кастомізації, конкретно для користувача.

Основним і вагоми мінусом всіх вище зазначених месенджерів являється те, що вони не підтримують роботу з додатковим програмним забезпеченням, і в основному перенасичені не потрібними функціями, які в деяких випадках, більше заважають користувачу.

В даному веб-додатку планується максимально оптимізувати роботу, на сам перед, користувача з інтерфейсом. Позбавившись від всього лишнього і максимально оптимізуючи систему, для швидкої і на сам перед зручної роботи стандартного користувача інтернет.

Не мало важну роль відіграє дизайн веб додатку. На сьогоднішній день, один із найважливіших параметрів, які стоїть на рівні з швидкістю і функціоналом є зовнішній вигляд веб додатка, який повинен бути виконаний у максимально зручному і зрозумілому вигляді для користувача.

Оптимізуючи веб додаток, можна добитися максимальної продуктивності, самого додатку, що на сьогоднішній час є дуже актуальним. При роботі з веб додатком, не малу роль, відіграє його можливість обробки запитів, і їх відправки. Швидкість додатку, його ключова перевага, перед іншими додатками. На сам перед безпосередньо запуск.

Основними недоліками, вище зазначених систем, на мою думку, є недостатня оптимізація додатку, що в свою чергу приносить дискомфорт користувачу, також вагомим недоліком можна вважати перенасичення дизайну сторонньою рекламою. Отже актуальною є задача яка б враховувала такі недоліки, та не втрачала при цьому переваг, такі як зручність у використанні, та стабільна робота.

Таким чином, розроблювана веб орієнтована система обміну текстовими повідомленнями має бути позбавлена існуючих систем, такі як:

- погана реалізація збереження конфіденційної інформації користувача;
- значно врізаний функціонал;
- відсутня можливість синхронізації з стороннім програмним забезпеченням;
- відсутність можливості кастомізації інтерфейса безпосередньо самим користувачем;
- не завжди вдалий дизайн, насамперед перенасиченість кольорами;
- не завжди працює синхронізація між девайсами, при прочитаних повідомленнях;

- відсутня інтеграція з різними пристроями користувачів і синхронізація;
- відновлення паролів, реєстрація, і зміна інформації користувача, відбувається з допомогою відправки SMS-повідомлення на телефон користувача.

Отже, існуючі інформаційні системи обміну повідомленнями в основному надають користувачам функції спілкування, знаходять інформацію, поширюють корисну інформацію та об'єднують великі групи людей із спільними інтересами.

1.3 Аналіз сучасних засобів створення програмного забезпечення

Для розробки веб орієнтованого додатку по обміну текстовими повідомленнями можна використати різноманітні технології, зокрема такі як .NET [19], Java та PHP та інші.

Microsoft .NET [18] – це програмна технологія, надана Microsoft, яку можна використовувати в якості платформи для створення звичайних програм і веб-додатків. В основному це продовження ідей і принципів, закладених в технології Java. Одна з ідей .NET - сумісність сервісів, написаних на різних мовах. Microsoft рекламує цю функцію як перевага .NET, але платформа Java має ті ж функції.

Кожна бібліотека (колекція) .NET має свою власну версію для вирішення конфліктів, які можуть виникнути між різними версіями колекції.

.NET - це крос-платформна технологія, яка тепер є варіантом платформи Microsoft Windows FreeBSD (проводиться Microsoft) і проекту Mono (на основі угоди між Microsoft і Novell) для Linux DotGNU [1].

Захист авторських прав застосовується до створення середовища загальномовного середовища виконання (CLR) для програм .NET. Багато компаній безкоштовно надають компілятори .NET на різних мовах.

.NET розділений на дві основні частини: Виконавча (в основному віртуальна машина) та інструменти розробки.

Середовище розробки .NET: є програми для розробки програм .NET. Ви також можете використовувати консольний компілятор в текстовому редакторі для розробки програм.

Як і технологія Java, середовище розробки .NET генерує байт-код для роботи у віртуальній машині. Рідна мова введення в .NET також називається CIL (загальна проміжна мова), MSIL (проміжна мова Microsoft) або коротше IL. Байт-код дозволяє отримувати крос-платформа на складеному рівні проекту (.NET: у формі збірок) замість рівня необробленого тексту, такого як C. Перед запуском збірки під час виконання (CLR) байт-код був перекладений (використовується лише вбудований компілятор JIT, щоб "влітати" в машинний код цільового процесора в часі.

Варто відзначити, що Microsoft також розробила один з найбільш ранніх JIT-компіляторів Java (в даний час Java використовує Sun HotSpot, який являє собою більш просунутий багаторівневий компіляцію). Новітня технологія динамічної компіляції дозволяє досягти того ж рівня продуктивності, що і традиційні «статичні» компілятори (наприклад, C++). Проблеми з продуктивністю зазвичай залежать від якості компілятора.

Платформа також включає технологію ASP.NET, яку можна використовувати для розширення системи обліку.

PHP [6] (PHP: Hypertext Preprocessor), раніше відомий як: Інструмент персональної домашньої сторінки, створений мовою сценаріїв для створення HTML-сторінок на стороні веб-сервера. PHP - одна з найпопулярніших мов у веб-розробці. Більшість постачальників послуг хостингу підтримують PHP. PHP - це проект з відкритим кодом.

Веб-сервер інтерпретує PHP як HTML-код і передає його клієнту. На відміну від мови сценаріїв JavaScript, браузер отримує повний HTML-код, тому користувач не може бачити PHP-код. Це дає перевагу в плані безпеки, але знижує інтерактивність сторінки. Однак ніхто не забороняє використовувати PHP для генерації коду JavaScript, який вже запущений на клієнті..

На платформі PHP, на відміну від ASP.NET, MySQL слід використовувати в якості бази даних замість ASP.NET MS SQL Server.

Хоча ці три платформи мають свої переваги та недоліки, платформа PHP була обрана, тому вона, як правило, дуже підходить для розробки програмного забезпечення, і в цьому випадку функціональність програми може бути розширена.

1.4 Постановка задачі та вимоги до розробки інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка веб орієнтовної системи обміну текстовими повідомленнями на платформі PHP, що виконує наступні основні функції:

- реєстрація користувачів та робота з ними (реєстрація і авторизація користувачів);
- робота з особистими даними користувачів (редагування даних профілю);
- робота з повідомленнями (відправлення, одержання, формування діалогів, переліку діалогів);
- робота з профілями інших користувачів (пошук, перегляд відкритої інформації).

Розділ 2

Проектування інформаційної системи

2.1 Функціональна структура та бізнес-процеси системи

При розробці веб орієнтованої системи обміну текстовими повідомленнями необхідно автоматизувати наступні бізнес-процеси:

- бізнес-процес «Робота з відкритою інформацією»;
- бізнес-процес «Робота з повідомленнями»;
- бізнес-процес «Робота з діалогами»;
- бізнес-процес «Робота з особистими даними»;

Бізнес-процес «Робота з відкритою інформацією». Даний бізнес-процес надає можливість користувачу отримувати інформацію про його співрозмовника. Зокрема, автоматизації підлягають наступні функції:

- 1) Виведення інформації про користувача.
- 2) Виведення інформації про діалог з користувачем.

Бізнес-процес «Робота з повідомленнями». Даний бізнес-процес надає можливість користувачу редагувати і видаляти надіслані ним повідомлення. Зокрема, автоматизації підлягають функції виведення, редагування інформації.

Бізнес-процес «Робота з діалогами». Даний бізнес-процес надає можливість користувачу переглядати і сортувати діалоги.

Бізнес-процес «Робота з особистими даними». Даний бізнес-процес надає можливість працівнику поліклініки працювати з особистими даними. Зокрема, автоматизації підлягають наступні функції:

- 1) реєстрація в системі;
- 2) авторизація в системі;
- 3) редагування даних профілю.

На початку роботи з системою користувачу виводиться стартова сторінка, де він бачить інформацію про авторів, системи та назву самого продукту. Звідси він може перейти на сторінку «Home», попередньо зареєструвавшись, де відображається інформація про користувача та діалоги які він веде. Також з

стартової сторінки користувач може залогінитись та зареєструватись. На рисунку 2.1 зображена UML-діаграма сайту, що зображує описані бізнес-процеси.

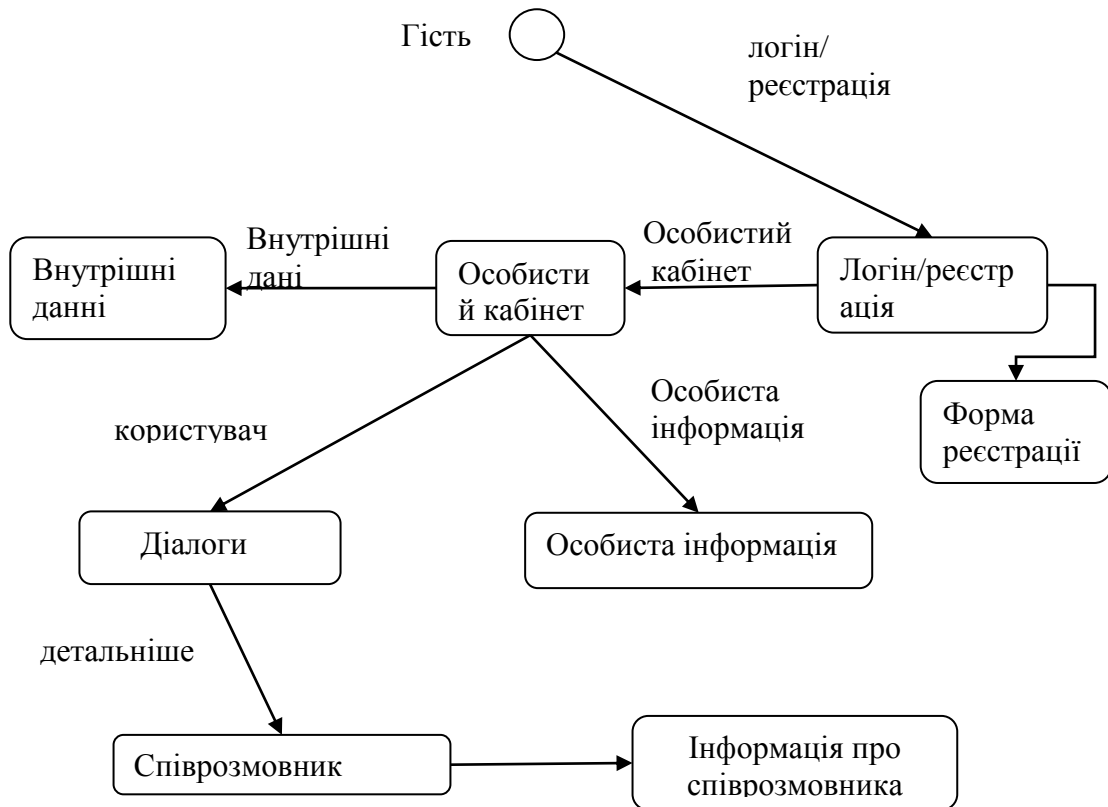


Рисунок 2.1 – UML-діаграма сайту.

Пройшовши авторизацію, користувач переходить на домашню сторінку, де бачить інформацію про діалоги. З цієї сторінки користувач може перейти на сторінку з більш детальною інформацією про діалог, на сторінку з особистими даними користувача.

На сторінці з більш детальною інформацією про діалоги користувач може продивлятися, редагувати та видаляти повідомлення, надіслані ним раніше, а також особисту інформацію співрозмовника.

На сторінці зі особистими даними користувача, він може продивлятися та редагувати його особисті дані.

На сторінці зі внутрішніми даними системи, користувач може продивлятися, редагувати внутрішні дані системи.

Основними бізнес-процесами даної системи, являються «Відправка повідомлення», «Збереження діалогів користувачів», «Збереження інформації про користувачів», «Збереження даних для авторизації в системі».

Бізнес-процес «Відправка повідомлення», відповідає безпосередньо за комунікацію між користувачами у самій системі обміну повідомленнями. Він також відповідає за прийом повідомлень від сторонніх користувачів і також, відправлення повідомлень іншим користувачам.

Бізнес-процес «Збереження діалогів користувачів», відповідає за збереження діалогів між користувачами, та також збереження повідомлень від невідомих користувачів, які залишились без відповіді.

Бізнес-процес «Збереження інформації про користувачів», відповідає за збереження інформації про користувача, яка була вказана при реєстрації користувача в системі.

Бізнес-процес «Збереження даних для авторизації в системі», відповідає за збереження інформації користувача при авторизації в системі, а саме логін і пароль.

Всі вище зазначені бізнес-процеси зображені на рисунку 2.2

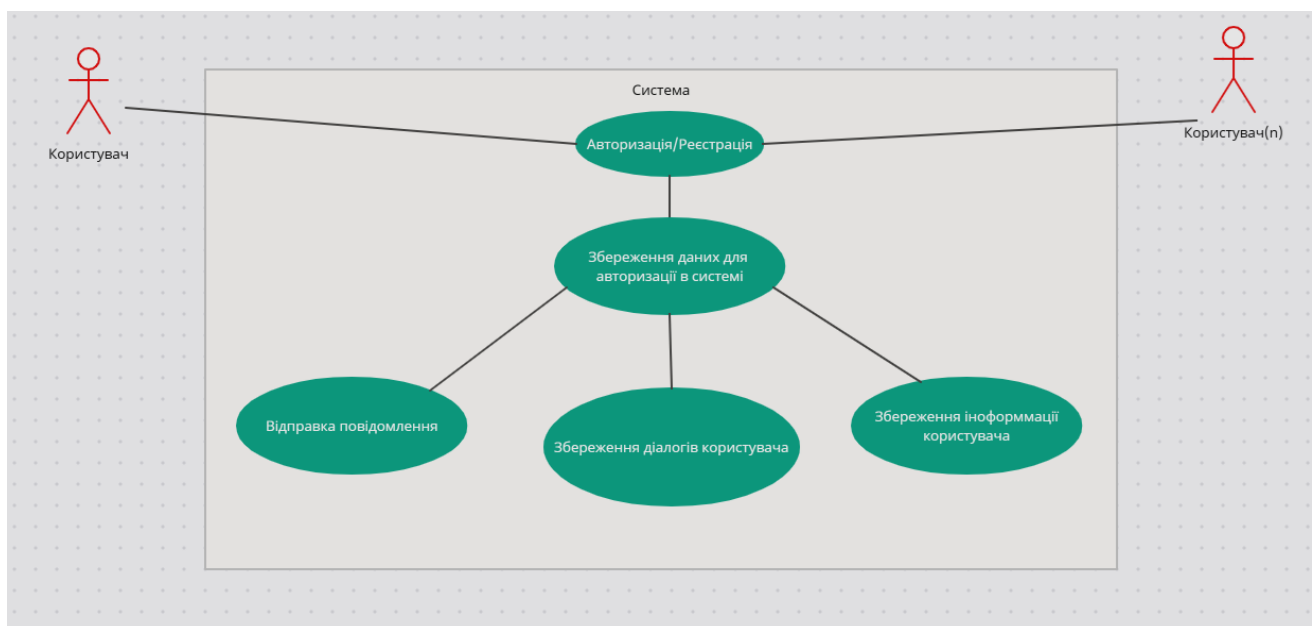


Рисунок 2.2 – UML-діаграма бізнес-процесів сайту.

Для збереження даних по розглянутих бізнес-процесах та послідовності дій користувача системи, необхідно використати БД.

2.2 Інформаційна структура системи

Для забезпечення збереження даних було розроблено БД, даталогічна модель якої зображена на рисунку 2.3.

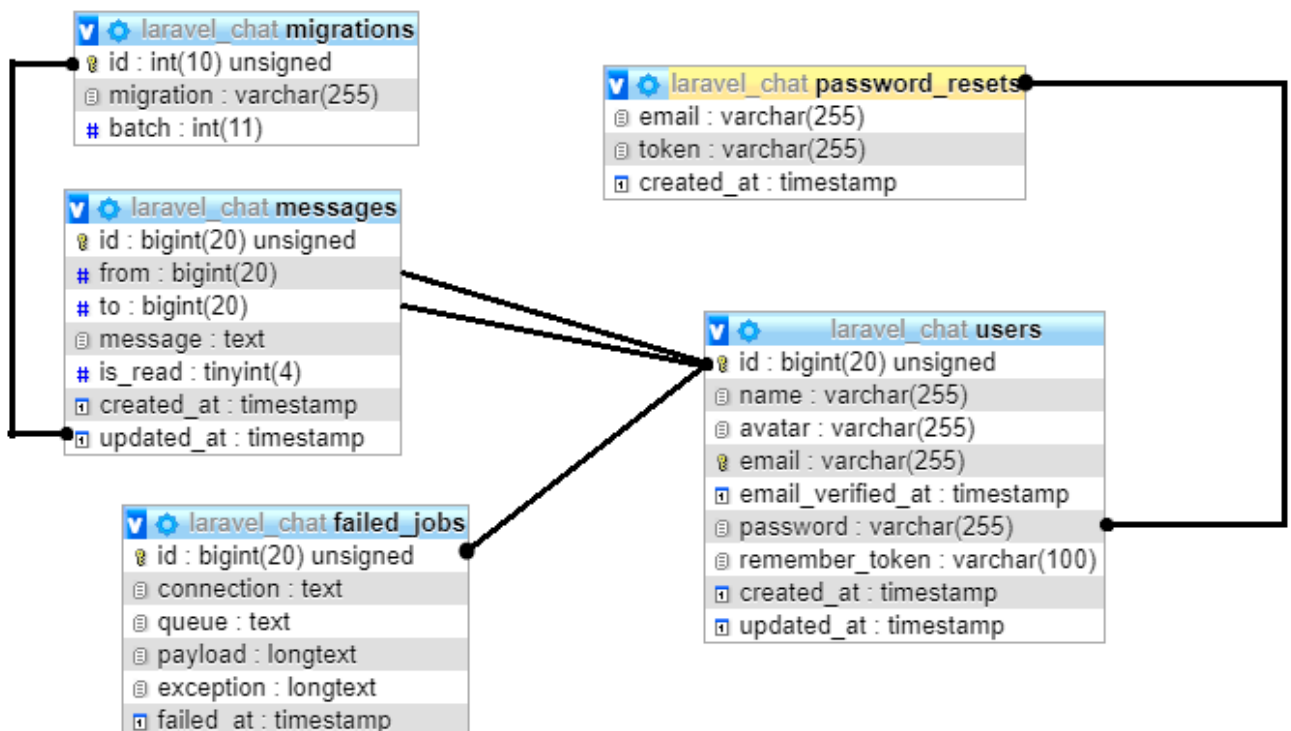


Рисунок 2.3 – Схема бази даних

Дана база даних знаходиться у 3 Нормальній формі. Головною таблицею є таблиця «users», для зберігання всієї особистої інформації про користувача у яку заносяться їхній Email та пароль.

Також добавлена таблиця «messages» з метою зберігання повідомлення, відісланих між користувачами, з можливістю редагування, видалення або зберігання для подальшого користування. Також у БД зберігається час і дата надіслання повідомлення від одного користувача, другому. А також зберігається всі зміни у вже надісланих повідомлення. Таблиця «messages» призначена для

занесення, редагування, видалення і збереження повідомлень між користувачами (таблиця 2.1).

Таблиця 2.1 – Атрибути таблиці «messages»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	from	int	ID користувача від якого прийшло повідомлення
3	to	int	ID користувача якому прийшло повідомлення
4	message	varchar	Текст повідомлення
5	is_read	int	Індикатор того чи прочитане повідомлення
6	created_at	datetime	Дата створення запису
7	updated_at	datetime	Дата оновлення запису

Всі невдалі спроби зберігання інформації про користувачів, або їх редагування, зберігається в таблиці «failed_jobs». Що надає можливість побачити всі невдалі спроби реєстрації користувачів, або ж невдалі спроби авторизації на сайт, що в свою чергу допомагає відслідкувати не добросовісних користувачів, при спробі взламу чужого акаунту. Таблиця «failed_jobs» призначена для занесення туди даних про невдалі спроби реєстрації чи авторизації (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «failed_jobs»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	connection	int	Інформація про те, чи відбулося з'єднання
3	queue	int	Послідовність дій(черга)
4	payload	varchar	Навантаження
5	exception	varchar	Виняток
6	failed_at	datetime	Дата оновлення запису

Також була додана таблиця «password_resets» для скидання паролів, і запису всіх дій над ним. Таблиця «password_resets» призначена для скидання пароля, та будь якої роботи над ним (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «password_resets»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	email	varchar	Email користувача
3	token	varchar	Токен користувача
4	create_at	datetime	Дата створення запису

Таблиця «users» призначена для додавання, редагування, зберігання і видалення всієї отриманої інформації про користувача, на даний момент (таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «users»

№	Назва атрибуту	Тип даних	Опис
1	id	int	Лічильник, унікальний
2	name	int	Зовнішній ключ до таблиці Імя
3	avatar	int	Зовнішній ключ до таблиці Зображення
4	email	int	Зовнішній ключ до таблиці Email
5	password	int	Пароль користувача
6	created_at	datetime	Дата створення запису
7	updated_at	datetime	Дата оновлення запису

Для даної системи не потрібна велика кількість бізнес-процесів, так як вона являється вузько спеціалізованою програмою, для вирішення конкретної проблеми, у даному випадку, це комунікація між користувачами всесвітньої мережі інтернету.

Робота з даною системою буде відбуватися за допомогою чотирьох таблиць, а саме: «users», «messages», «password_resets» і «failed_jobs». Кожна із цих таблиць виконує свою, важливу роль у даній системі.

Так для бізнес-процесу «Збереження інформації про користувачів» використовувалась таблиця «users», яка зберігає всю наведену інформацію про

користувачів, у базі даних, для роботи з даною інформацією в майбутньому, з можливістю її видалення, обробки або ж подальшого збереження її в базі даних.

Таблиця «messages» призначена для роботи з бізнес-процесом «Збереження діалогів користувачів», яка в свою чергу відповідає за додавання, редагування, видалення і збереження повідомлень між користувачами.

За бізнес-процес «Збереження даних для авторизації в системі» відповідає таблиця «failed_jobs» призначена для занесення туди даних про невдалі спроби реєстрації чи авторизації. Таблиця «password_resets» призначена для скидання пароля, та будь якої роботи над ним.

Отже, зазначений обсяг таблиць є достатнім для обробки всіх необхідних даних для роботи з месенджером.

2.3 Вибір засобів розробки інформаційної системи

Для розробки веб орієнтованого додатку по обміну текстовими повідомленнями на платформі PHP слід обрати фреймворк Laravel який має відкритий вихідний код, і має різні підходи до реалізації роботи з різними типами бази даних. Мовою програмування була вибрана PHP, що є так званою мовою сценаріїв, яка використовує сторінки для веб-сервера. Для розробки цього додатку була вибрана БД MySQL. Ця система знаходиться у вільному доступі і найкраще підходить для роботи з великими базами даних, також варто зазначити її швидкодію.

2.3.1 Вибір мови програмування

PHP [7] (PHP: препроцесор гіпертексту, раніше відомий як «інструменти для персональної домашньої сторінки») – мова програмування сценаріїв, який використовується для створення HTML-сторінок на стороні веб-сервера. PHP - один з найбільш часто використовуваних мов в веб-розробці. Більшість

провайдерів хостингу підтримують PHP. PHP – проект з відкритим вихідним кодом.

Варто зазначити простоту використання даної мови програмування. Така мова як PHP зазвичай вбудована безпосередньо в html-код сторінок, які легко можуть оброблятися PHP-інтерпретатором. PHP містить в собі величезну різноманітність і кількість різних популярних функцій, що позбавляє нас такої необхідності, як писати складні та багаторядкові скрипти для виконання частіш за все простого завдання [7] . На сьогоднішній день, головне для розробника – швидко і правильно вибрати функцію або метод відповідно до конкретного поставленого до нього завдання. Також варто зауважити, що не потрібно встановлювати конкретні бібліотеки, вказувати спеціальні параметри початкової компіляції.

Також варто звернути увагу на засоби безпеки системного рівня даної мови програмування. У своєму функціоналі PHP має можливість налаштування так, щоб вона може забезпечувати максимальну свободу дій і водночас безпеку високого рівня. PHP може вести роботу в безпечному режимі, який в свою чергу обмежує можливості його застосування, PHP користувачами. Прикладом може слугувати: максимальний час виконання для роботи та використання об'єму пам'яті. В такому випадку засоби безпеки рівня програми, виконують основну роль в роботі програми. PHP включає в себе надійні механізми та функції шифрування. PHP також має можливість підключення з додатками інших користувачів, що в свою чергу дозволяє легко і доступно інтегрувати отриманий програмний код із його захищеними технологіями включаючи електронної комерції. Одним із мінусів є те, що вихідний код PHP, на даний момент, не можна переглянути у будь якому браузері, оскільки його обробка відбувається безпосередньо на сервері.

Стратегія відкритого коду і розповсюдження початкового коду програмного продукту в масах, безсумнівно створило великий вплив на велику кількість проектів. Для розробників відіграла велику роль стратегія відкритого коду, що значно спрощує роботу з програмним кодом. Тому сказане часто

відноситься і до самої теми створення PHP, в такому випадку коли, підтримка бази користувачів, які налічуються по всьому світу, виявилася одним із найважливіших чинників в розвитку мови програмування PHP [7]. Остаточний перехід на відкритий код і безкоштовне розподілення початкових програмних кодів PHP надало велику допомогу користувачам. Також, всі користувачі PHP в у світі є так званою колективною службою підтримки, і в багатьох випадках популярні електронні конференції, в яких можна знайти вирішення проблеми на складні задачі, які потребують додаткової уваги.

Варто зазначити що ефективність, на сьогоднішній день, є одним із важливих чинників при програмуванні для конкретних середовищ роботи, які в свою чергу розраховані на велику кількість користувачів, до яких належить і веб-розробка. Важливою перевагою в сторону PHP є те [9], що дана мова програмування належить до інтерпретованих. Дана мова програмування дозволяє обробляти створені користувачем сценарії з достатньо високою швидкістю обробки. Обробка інформації, PHP-сценарії, особливо не дуже великих розмірів, обробляються швидше ніж аналогічні їм програми, що були створені користувачем. Проте, для того щоб не робити програмні шаблони PHP, виконувані ними файли, отримані за допомогою компіляції програмного коду, будуть працювати значно швидше, що пришвидшує роботу програмного забезпечення. Варто зазначити що продуктивність PHP коду цілком достатня для створення цілком автоматизованих і оптимізованих серйозних веб-додатків. Вище вказана мова програмування дозволяє працювати з великим об'ємом користувацького сценарію з достатньо високою швидкістю обробки даних.

Веб-сервер інтерпретує PHP як HTML, а потім передає HTML клієнту. На відміну від мови сценаріїв JavaScript, користувач не побачить PHP-код, оскільки браузер отримає готовий html-код. Це перевага в безпеці, але це зменшить інтерактивність сторінки. Але ніхто не забороняє використовувати PHP для генерації коду JavaScript, який вже виконується на стороні клієнта.

2.3.2 Вибір фреймворку

Laravel [2] – це безкоштовний PHP-фреймворк з відкритим вихідним кодом, створений Т. Отвеллом, метою якого є розробка веб-додатків на основі шаблонів модель-уявлення-контролер (MVC). Деякі з функцій Laravel - це модульна пакувальна система зі спеціальним менеджером залежностей Composer, різні способи доступу до реляційних баз даних, утиліти, які допомагають розгорнути і обслуговувати додатки, а також акцент на синтаксичному цукрі.

Одним із вагомих переваг вибраного фреймворку є те, що міграції бази даних відбуваються з участю користувача, що є дуже корисним для будь-якого за величезною проекту, тому особливо для великих проектів з кількома передовими розробниками які дозволяють встановлювати останню версію бази даних для всіх розробників [10]. Вибраний фреймворк Laravel є потрібен лише для однієї команди в програмному коді. Також у даному фреймворку доступний конструктор таблиць, що відкриває можливість швидкого редагування даних, які були збережені в базі даних.

Розробка програмного продукту на базі фреймворка Laravel має ряд переваг, такі як досить широкий і різносторонній функціонал і вибір модулів. В даному фреймворку можна використовувати масштабні завдання, незалежно від масштабності і складності, також є можливість створення багаторічного веб-додатку . І що найважливіше у функціоналі даного фреймворку є те, що він дозволяє грамотно і швидко знаходити вирішення нестандартних проблем.

Також надається можливість тестування, додавання і оновлення заданого змісту в роботу веб-додатку на створеній користувачем версії. Таким чином за рахунок внутрішньої підтримки різних версій веб-додатку. Зручна і легка для розуміння админ-панель [10]. Демонструється великий спектор вбудованих можливостей редагувати правки в адміністративну панель і вибирати найбільш вигідний і зручний варіант управління даними в БД.

Також варто зазначити що, Laravel сприяє розробникам веб-додатків і для цього спростили процес розробки. Використавши простоту коду, за допомогою якої виконується багаторазовість використання коду. Цей фреймворк один із небагатьох, який має великий потенціал в майбутньому, щоб зробити її прогресивною платформою для розробки веб-додатків та онлайн-платформ.

Laravel встановлюється з архітектурною схемою моделі-перегляду-контролера. Ця система надзвичайно проста у використанні і, тому, пропонує великий діапазон зручних способів створення масштабних або малих веб-додатків. За допомогою цього, розробники мають можливість організувати масштабні проекти з великою кількістю файлами для поліпшення роботи.

У випадку якщо вам потрібно провести роботи над великим об'ємом роботи, користувачу доведеться працювати з справу великою кількістю програмного коду, який є не дуже стандартним кодом. Починаючи використовувати систему MVC користувач може спростити загальну структуру програмування та полегшити користувачу роботу з даними. Також даний фреймворк спрощує роботу з пошуком файлів, у своїх локальних папках під час пошуку над великою кількістю файлів. За допомогою даного фреймворка користувач може контролювати всі процеси на проекті, так ваша веб-додаток компілюється в реальному часі, так користувач може вдосконалювати свій веб-додаток компонентним підходом.

Логіка Laravel це модуль розподіленого веб-додатку, оголошений або за допомогою контролерів, або маршрутів. Таким чином зворотне з'єднання пов'язує між собою генерування веб-додатком посилання і зв'язки, які в свою чергу дозволяють редагувати основні автоматичні оновлення які пов'язані з посилань. Під час роботи з посиланнями іменованих з'єднань Laravel автоматично створює кінцеві URL.

Станом на березень 2021 року, Laravel вважається одним з найпопулярніших PHP фреймворком, разом з Symfony2, Nette, CodeIgniter, Yii2 й іншими фреймворками [10]. Серцевий код Laravel'a розміщується на GitHub і ліцензований відповідно до умов MIT License.

2.3.3 Вибір редактора програмного коду

Для цієї задачі було вибрано редактор коду Visual Studio Code, який на даний момент є одним із найпопулярніших редакторів програмного коду. Цей редактор заслужив свою популярність, оскільки відрізняється від усіх своєю простотою, багатofункціональністю, зручністю і доступністю. Він надає можливість встановлення плагінів для роботи, що значно спрощують роботу у написанні коду користувачем, що в свою чергу сприяє швидшій і продуктивнішій роботі за написанням програмного коду. Visual Studio Code був представлений компанією Майкрософт на весні 2015 року на конференції Build 2015. Також в редакторі встановлений додаток для зручної роботи з Git. Також даний редактор підтримує переважну більшість мов програмування, що в і робить його таким популярним на просторах інтернету.

VSCode підтримує виконання завдання через палітру команд. Як завдання можуть йти, наприклад, пакетний менеджер скрипти. VSCode автоматично визначає такі команди і надає їх список при виконанні команд. Після вибору завдання зі списку в перший раз інформація по ньому пропишеться в «tasks.json» у вашій директорії. Наступного разу ж буде запускатися відразу. Запущеному завданню виділяється окремий термінал, який відкривається після запуску, а список запущених завдань відображається в нижньому барі. При натисканні на кнопку виведеться список завдань, вибране завдання перекине на відповідний термінал.

Незважаючи на досить велику популярність програми, її функціонал досі залишається не до кінця очевидним всім користувачам, які її використовують, через це багато користувачів вважають, що функціонал є занадто малий і кращим буде вибір аналогічного продукту конкурента, ніж VS Code [11]. Щоб вирішити всі поставлені завдання і проблеми, що виникають у кожного користувача при роботі з редактором коду Visual Studio Code, тому цей редактор потребує особливої уваги до ознайомлення із всім доступним функціоналом і

наявними в ньому плагінами, тому доречним буде ознайомитися з докладною інструкцією по налаштуванню даного редактора коду.

2.3.4 Вибір СКБД

Відповідно до переваг і недоліків різноманітних СКБД було обрано MySQL, так як вона підходить під платформу PHP і задовольняє всі потреби даної задачі, по реалізації веб орієнтованого додатку по обміну текстовими повідомленнями.

MySQL [23] – вільна система керування реляційними базами даних. ТСХ розробив MySQL для покращення швидкості обробки великих баз даних. Створить цю систему управління базами даних із відкритим кодом (СКБД) як альтернативу комерційним системам. MySQL з самого початку був дуже схожий на MySQL, але з часом він розширювався, і тепер MySQL став однією з найпоширеніших систем управління базами даних. В основному він використовується для створення динамічних веб-сторінок, оскільки має чудову підтримку різних мов програмування. В першу чергу вона використовується для створення динамічний веб-сторінок, тому як має хорошу підтримку з різних мов програмування. MySQL характеризується великою швидкістю та стійкою і що не менш важно простою у використанні будовою, що добре впливає на обробку великої інформації. Також ця система проста у встановленні і підключенні, підтримує необмежену кількість користувачів, які одночасно можуть працювати за базою даних, необмежена кількість рядків у таблицях. Одним із найважливіших аспектів є те, що дана MySQL має просто і ефективну систему безпеки.

Отже, для роботи з даним проектом було вибрано мову програмування PHP і програмний каркас під назвою Laravel. Також розглядалась можливість вибору іншого програмного забезпечення і проводилися тести для оптимального вибору програмного забезпечення, для оптимального вирішення поставленої задачі. Розглядався варіант вибору сучасної мови програмування JavaScript, но

при роботі з базами даних MySQL, робота з даною мовою програмування не була коректною, тому було вирішено зупинитися на мові програмування PHP, яка на даний момент також швидко розвивається, і має добре програмне забезпечення. На даний момент мова програмування PHP є досить популярною, і оновлюється дуже часто, що надає можливість використовувати новий функціонал у вже існуючі веб додатки, або добавляти нові. Тому було вирішено використовувати одержану комбінацію засобів розробки.

Розділ 3

Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення складових системи

При розробці веб орієнтованої системи обміну текстовими повідомленнями використовується фреймворк Laravel, який безпосередньо реалізує шаблон дизайну системи MVC. Вся мета цього шаблону – безпосередньо розділити програмне забезпечення на певні моделі, контролери і подання. Модель буде реалізовувати бізнес-логіку програми. Тому в додатку велика кількість модулів. Це уявлення також містить висновок системних даних. У цьому випадку контролер реалізує зв'язок між моделлю і представленням, як це зображено рисунку 3.1.

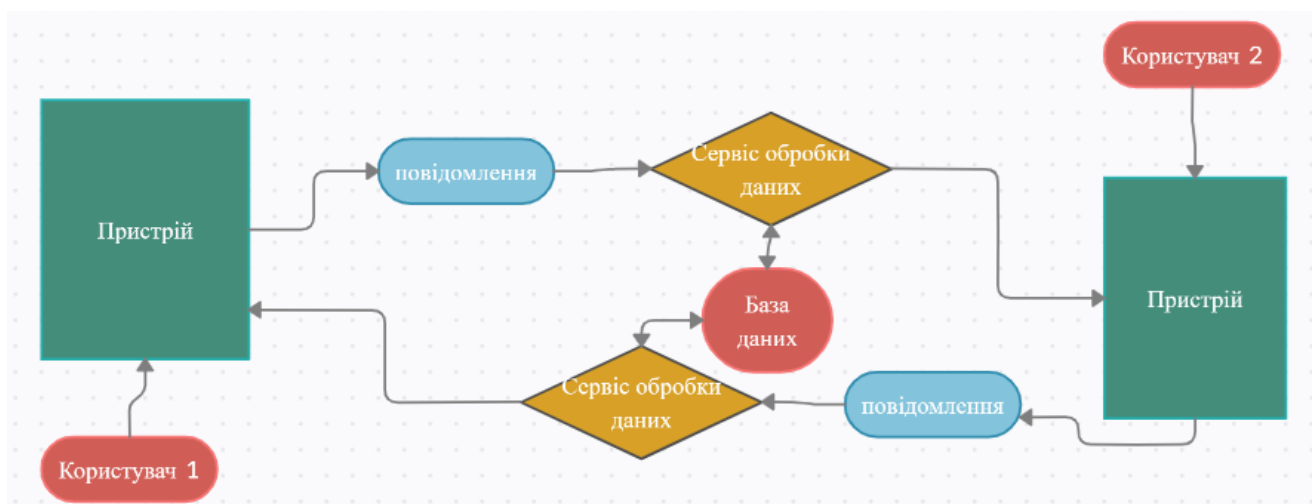


Рисунок 3.1 – Структура веб орієнтованої системи обміну текстовими повідомленнями

В даному проекті присутні наступні моделі (рисунок 3.2):

- User – є моделлю користувача, яка на пряму відповідає за бізнес логіку, пов'язану з всією особистою інформацією про даного користувача.
- Register - яке відповідає за реєстрацію користувача в системі.
- Login – модель яка відповідає за авторизацію користувача в системі.

– `PersonalData` - Відповідає за відображення особистих даних користувача.

– `Welcome` - Відповідає за відображення стартової сторінки.

– `InnerData` - Відповідає за відображення внутрішніх даних системи.

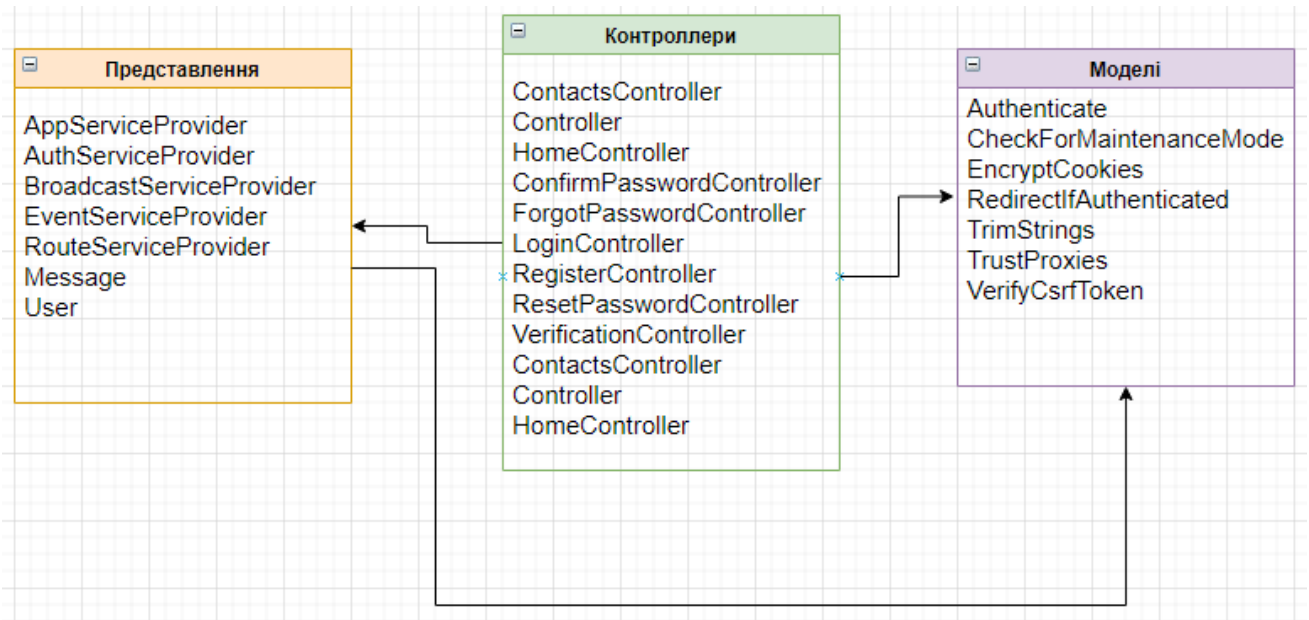


Рисунок 3.2 – Діаграма класів веб орієнтованої системи обміну текстовими повідомленнями

Також у системі є наступні контролери:

– `LoginController` - Відповідає за логінування.

– `RegisterController` - Відповідає за реєстрацію.

– `PersonalDataController` - Відповідає за особисті дані користувача.

– `WelcomeController` - Відповідає за початкову (стартову) сторінку.

3.2 Особливості реалізації складових системи

Основна складність у створенні програми для надсилання повідомлень полягає в архітектурному дизайні. Структура програми повинна бути розроблена таким чином, щоб можна було легко додавати нові функції.

Розробляючи веб орієнтований додаток по обміну текстовими повідомленнями, розробнику потрібно не лише закласти основу, яка вже існує в інших додатках, але йому також потрібно визначити майбутні вимоги.

Більшість методів проектування і розробки архітектури спрощені до модульних систем. Але модульність різна, сам модуль може бути величезним і цілісним.

При проектуванні архітектури модулі повинні бути розділені на автономні рівні. Потім об'єкти бізнес-логіки, представництва та даних відокремлюються і можуть змінюватися незалежно один від одного. Незалежно від розміру системи, цей метод може зберігати свою гнучкість, масштабованість (розширену кількість функцій) та тестування.

Одною з найважливіших моделей в даному проекті являється API – ядро програми для обміну повідомленнями в реальному часі. Він відповідає за встановлення з'єднання між клієнтом і сервером для аутентифікації, обробки подій, маршрутизації повідомлень і інших цілей. Джерело подій зберігає стан об'єкта, тобто чату, як послідовність подій. Додаток зберігає події в журналі подій, який представляє собою базу даних подій, і в цьому журналі є API для додавання і отримання подій сутності.

В процесі роботи елементів даного механізму API організовують багаторівневу сукупність ієрархій. При цьому компоненти отримують подібну структуру.

У деяких користувачів може бути велика кількість подій. Щоб оптимізувати завантаження додатка, API може час від часу отримувати знімки стану об'єкта. Щоб відновити поточний стан, додаток буде шукати останній знімок і події, що відбулися з моменту створення знімка. Робота модуля API зображена на рисунку 3.3.

Одним із не менш важливих модулів є авторизація та реєстрація користувачів у цій програмі. Однією з функцій цього модуля є збереження всіх необхідних функцій, але в той же час не втрачається простота реєстрації та цього дозволу.

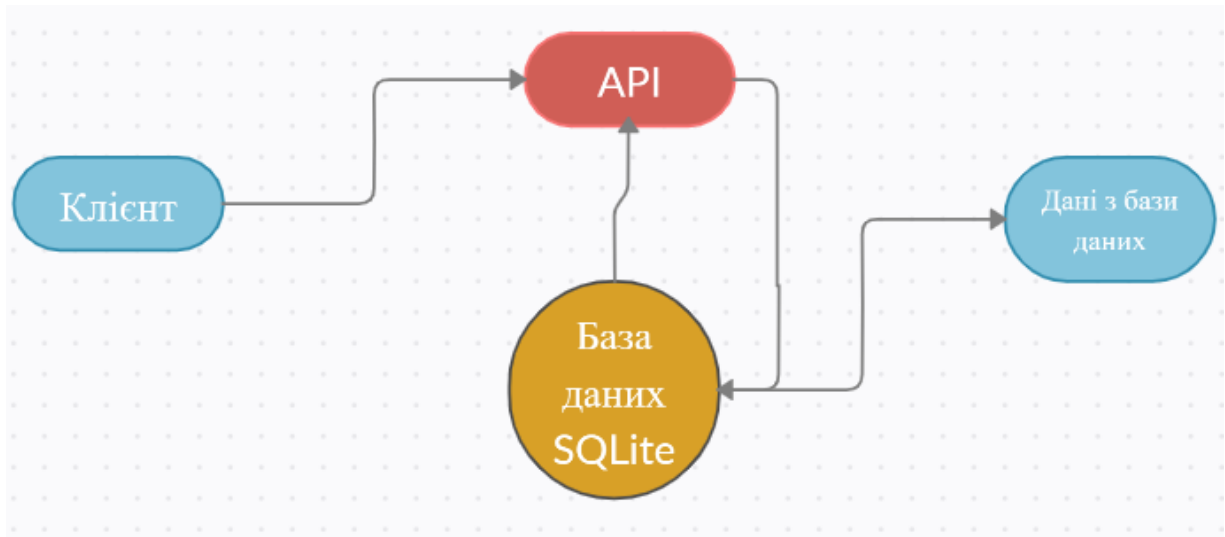


Рисунок 3.3 – Структура функціонування системи API

Реалізація обміну текстовими повідомленнями, була реалізована додавання для кожного користувача індивідуальний ключ, за яким проходив обмін повідомленнями. Це було реалізовано наступним програмним кодом:

```

public function getMessagesFor($id)
{

    Message::where('from', $id)->where('to', auth()->id()->update(['read' => true]));

    $messages = Message::where(function($q) use ($id) {
        $q->where('from', auth()->id());
        $q->where('to', $id);
    }->orWhere(function($q) use ($id) {
        $q->where('from', $id);
        $q->where('to', auth()->id());
    })
    ->get();

    return response()->json($messages);
}
  
```

```

        return response()->json($messages);
    }

    public function send(Request $request)
    {
        $message = Message::create([
            'from' => auth()->id(),
            'to' => $request->contact_id,
            'text' => $request->text
        ]);
    }

```

Структура Laravel забезпечує модуль для обробки аутентифікації користувачів. Файл конфігурації аутентифікації знаходиться в `config / auth.php`. Програмний код даного елемента:

```

protected function registerAuthenticator()
{
    $this->app->singleton('auth', function ($app) {

        $app['auth.loaded'] = true;

        return new AuthManager($app);
    });

    $this->app->singleton('auth.driver', function ($app) {
        return $app['auth']->guard();
    });
}

```

Він містить деякі докладні параметри для налаштування поведінки служби автентифікації. За допомогою цього модуля користувач може отримати форму реєстрації зображену на рисунку 3.4

Метод `validator ()` класу `RegisterController` містить правила перевірки введення даних для нових користувачів додатка, а метод `create ()` цього класу

відповідає за створення нових записів App \ User в вашій базі даних за допомогою Eloquent ORM. Ви можете змінити кожен з цих методів, як забажаєте. Даний метод був реалізований так:

```

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\User
 */

```

Метод attempt () приймає масив пар ключ / значення як перший аргумент. Значення масиву будуть використані для пошуку користувача в таблиці бази даних. Так користувач буде отримано за значенням стовпця email.

```

var result = attempt(function() {
    return Function(importsKeys, sourceURL + 'return ' + source)
        .apply(undefined, importsValues);
});

```

Рисунок 3.4 – Реєстрація користувача в базу даних

Якщо користувач буде знайдений, хеширувати пароль, збережений в базі даних, буде порівнюватися з значенням password, переданим в метод через масив. Якщо два хешировані пароля співпадуть, то для користувача буде запущена повторна авторизація. Програмний код:

```
public function attempt(array $credentials = [], $remember = false)
{
    $this->fireAttemptEvent($credentials, $remember);

    $this->lastAttempted = $user = $this->provider->retrieveByCredentials($credentials);

    if ($this->hasValidCredentials($user, $credentials)) {
        $this->login($user, $remember);

        return true;
    }

    $this->fireFailedEvent($user, $credentials);

    return false;
}
```

Також одним із головних модулів, який в свою чергу відіграє не менш важливу роль ніж всі вище вказані, це модуль стартової сторінки. На стартовій сторінці, відображається вся основна інформація про продукт.

Для того щоб всі нові повідомлення, які були надіслані користувачу, були позначені як прочитані, була реалізована наступна функція:

```
Message::where(['from' => $user_id, 'to' => $my_id])->update(['is_read' => 1]);

$messages = Message::where(function ($query) use ($user_id, $my_id) {
    $query->where('from', $user_id)->where('to', $my_id);
})->orWhere(function ($query) use ($user_id, $my_id) {
    $query->where('from', $my_id)->where('to', $user_id);
})->get();

return view('messages.index', ['messages' => $messages]);
}
```

Для переходу між сторінками було встановлено модуль роутенга, який зображений наступним кодом:

```
Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
Route::get('/message/{id}', 'HomeController@getMessage')->name('message');
Route::post('message', 'HomeController@sendMessage');
```

При запуску додатку, відображається стартова сторінка яка зображена на рисунку 3.5.

Наступним важливим модулем, буде, модуль збереженні діалогів і користувачів, з якими була здійснена взаємодія. Відображаються всі наявні діалоги і список користувачів, з якими вони велися. Всі користувачі вказані в списку зліва.

Форматування даних для надсилання на базу даних, було реалізовано наступним чином:

```
$factory->define(User::class, function (Faker $faker) {
    return [
        'name' => $faker->name,
        'avatar' => 'https://via.placeholder.com/150',
        'email' => $faker->unique()->safeEmail,
        'email_verified_at' => now(),
        'password' =>
'$2y$10$jWlKgEdG2oruR4UJ1SFyZOs9i.TTG4KEMT2VVH7WSpMD/jtMllrpm'
        , // 123456
        'remember_token' => Str::random(10),
    ];
});

$factory->define(Message::class, function (Faker $faker) {
    do {
        $from = rand(1, 30);
        $to = rand(1, 30);
        $is_read = rand(0, 1);
    } while ($from === $to);
});
```

Сторінка особистих даних, та інформацію про всі наявні діалоги із всіма наявними користувачами зображена на наступному рисунку. Тому було вирішено зібрати всю інформацію про користувача в окремий файл, який все форматує в потрібний для БД формат. Це було реалізовано наступним кодом:

```
use Notifiable;
* @var array
*/
protected $fillable = [
    'name', 'avatar', 'email', 'password',
];

*
* @var array
*/
protected $hidden = [
    'password', 'remember_token',
];

* @var array
*/
protected $casts = [
    'email_verified_at' => 'datetime',
];
```



Рисунок 3.5 – Стартова сторінка

Результат виконання даного модуля зображений на рисунку 3.6

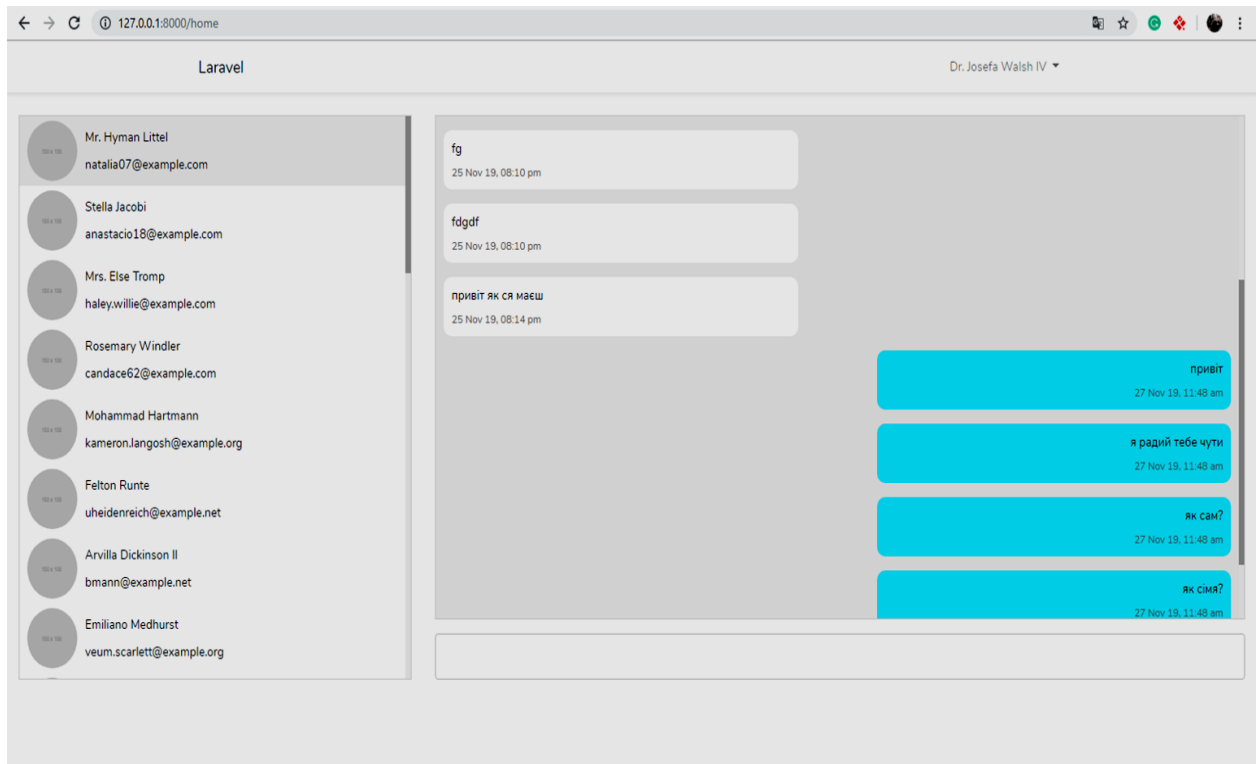


Рисунок 3.6 – Особиста сторінка користувача

Структура програми розроблена таким чином, щоб можна було легко додавати нові функції.

3.3 Тестування інформаційної системи

Для коректної роботи додатку, були проведені тест кейси, які показали надійність системи і її стійкість до не коректних дій. Для прикладу були використані стандартні випадки не коректної спроби використання даного програмного продукту. У першому тесті (Таблиця 3.1) була спроба авторизації не зареєстрованого користувача в системі, на даний тест, програма повинна видати попередження «Цього користувача немає зареєстрованого в системі» зображене на рисунку 3.7.

Рисунок 3.7 – Попереджувальне повідомлення

Таблиця 3.1 – Тест-кейс АТ0001

Тест-кейс ID: АТ0001	Пріоритет: 3	Створено: 08.05.2021.
Назва: Перевірка логіну заблокованого користувача		
Вхідні дані: Логін = «annsidsabelle25@example.org», Пароль = «123456»		
Кроки	Очікуваний результат	
<p><i>Передумова:</i> користувач з електронною поштою «annsidsabelle25@example.org» не повинен пройти реєстрацію, так як введений ним символ не підходить</p> <ol style="list-style-type: none"> 1. Перейти на веб орієнтований додаток 2. Вказати пошту і пароль 3. Нажати кнопку «Логін» 4. Побачити повідомлення про помилку 5. Дочекатися зникнення помилки 6. Ввести валідне значення електронної пошти «annabelle25@example.org» 7. Нажати «Логін» 8. Отримані результати порівняти 	<p>Побачити помилку «Частина перед «@» не може містити символ «i»»</p> <p>Закриття сторінки авторизації, перехід на головну сторінку.</p>	
Результат виконання тест-кейсу: пройдено успішно		

Наступним кейсом буде перевірена валідація. Користувач повинен вказувати валідну пошту. Пошта повинна бути вказана на латині, одним транслітом. При невірному вводі електронної адреси, система видасть

попередження користувачу, про не валідну пошту і підкаже в чому проблема, що зображено на рисунок 3.8.

Рисунок 3.8 – Валідація пошти

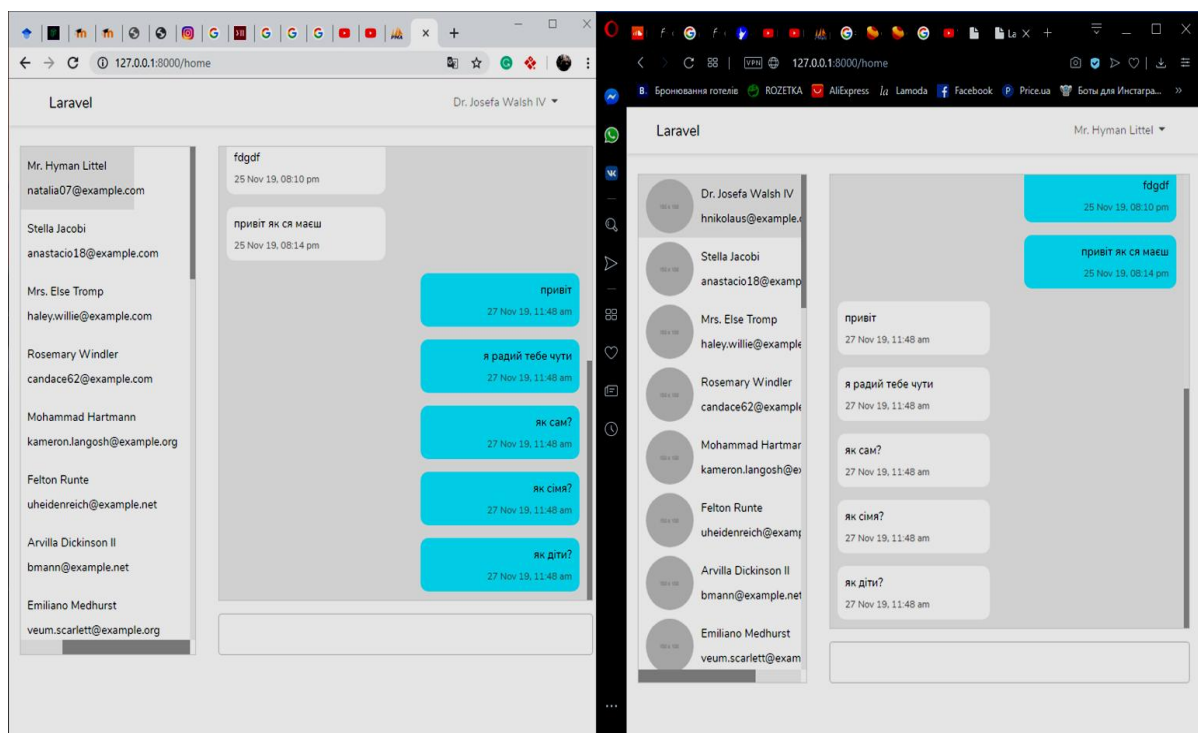


Рисунок 3.9 – Діалог між двома користувачами

Третім тест кейсом буде наявна робота веб орієнтованого додатку з різними користувачами між собою, в плані обміну текстовими повідомленнями і наявною роботою додатка в даній ситуації. Для цього тест кейсу, було зареєстровані два користувачі, які після їх реєстрації були авторизовані в веб

орієнтованому додатку. Щоб їх робота була наявна, було використано два браузера. Після їхньої авторизації, із одної сторінки користувача, на другу було надіслано повідомлення. Після відкриття діалогу між користувачами, було зображено відіслане повідомлення, що свідчить про відповідну роботу додатку, що було зображено на рисунку 3.9.

Отже, можна вважати, що діалог між двома користувачами відбувається і все працює коректно.

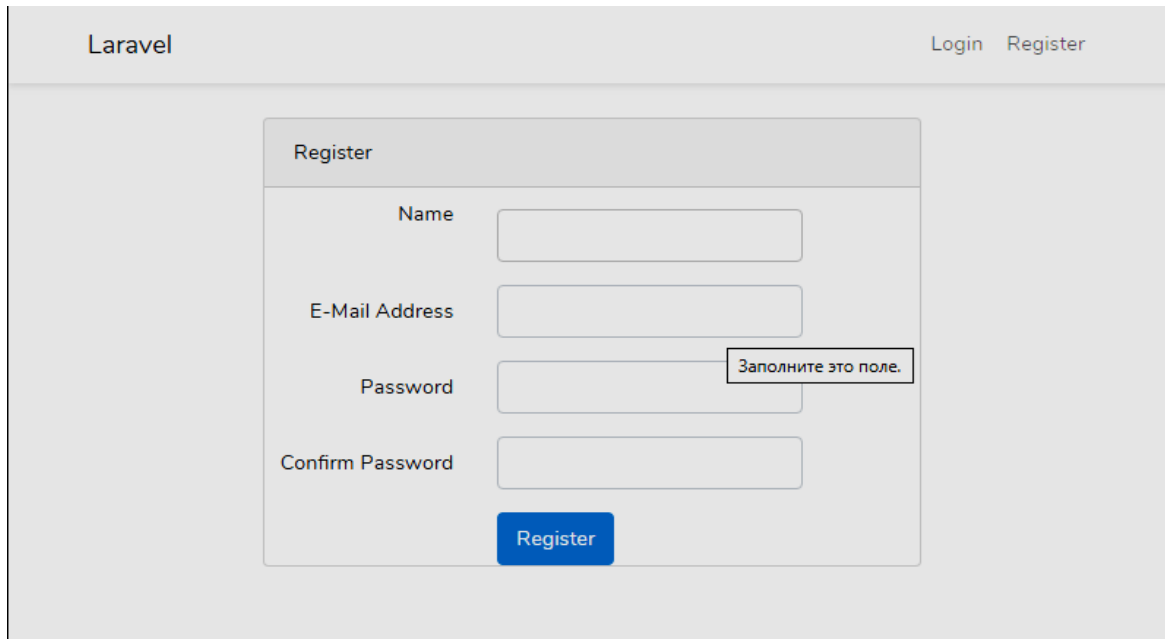
3.4 Інструкція користувача

Перше, що бачить користувач, зайшовши на сайт – це стартова сторінка, зображена на рисунку 3.11.



Рисунок 3.11 – Стартова сторінка

Далі користувач може зайти на вже існуючий опубліковий запис, або зареєструвати, якщо такого немає. Ця сторінка зображена на рисунку 3.12. Пройшовши авторизацію, користувач потрапляє на домашню сторінку, зображену на рисунку 3.13.



The screenshot shows the Laravel registration page. At the top left is the word "Laravel" and at the top right are links for "Login" and "Register". The main content is a "Register" form with the following fields: "Name", "E-Mail Address", "Password", and "Confirm Password". Each field has a corresponding input box. A tooltip with the text "Заполните это поле." (Fill in this field.) is positioned over the "Password" input box. Below the form is a blue "Register" button.

Рисунок 3.12 – Реєстрації користувачів

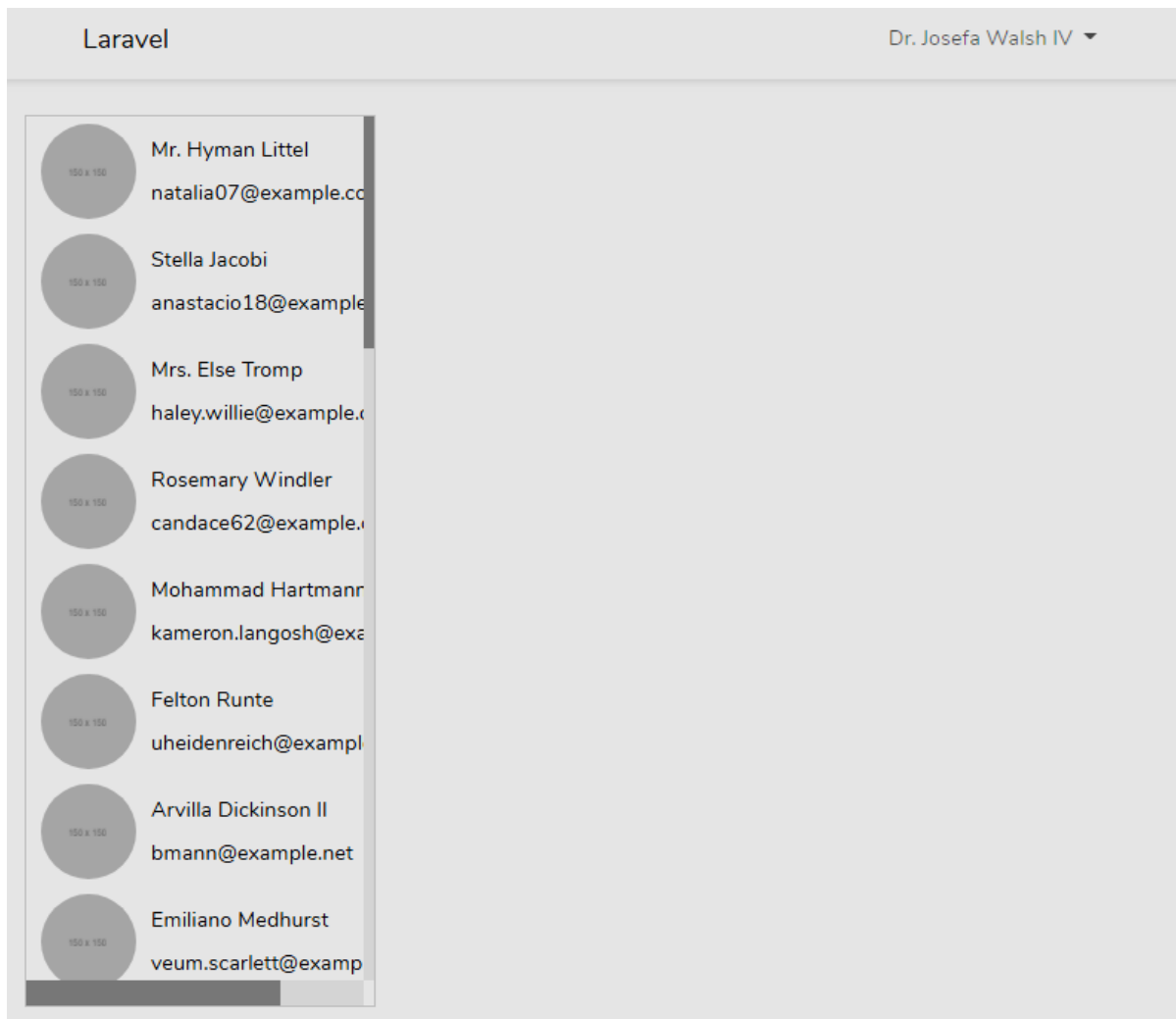


Рисунок 3.13 – Домашня сторінка

Отже, користування сайтом є інтуїтивно зрозумілим та не вимагає додаткових знань, умінь та навичок, окрім необхідних для звичайного користування комп'ютером.

Для роботи з серверною частиною продукту, знадобиться інсталяція додаткового програмного забезпечення, у вигляді Open Server[11], який зображено на рисунку 3.14. Дана інсталяція потрібна для підключення до проекту бази даних, яка відіграє в ньому велику роль. В базі даних зберігається вся інформація про користувачів, їхні діалоги і параметри авторизації. Тому для коректної роботи веб орієнтованого додатку, пропонується інсталювати даний додаток, перед початком роботи з сайтом.

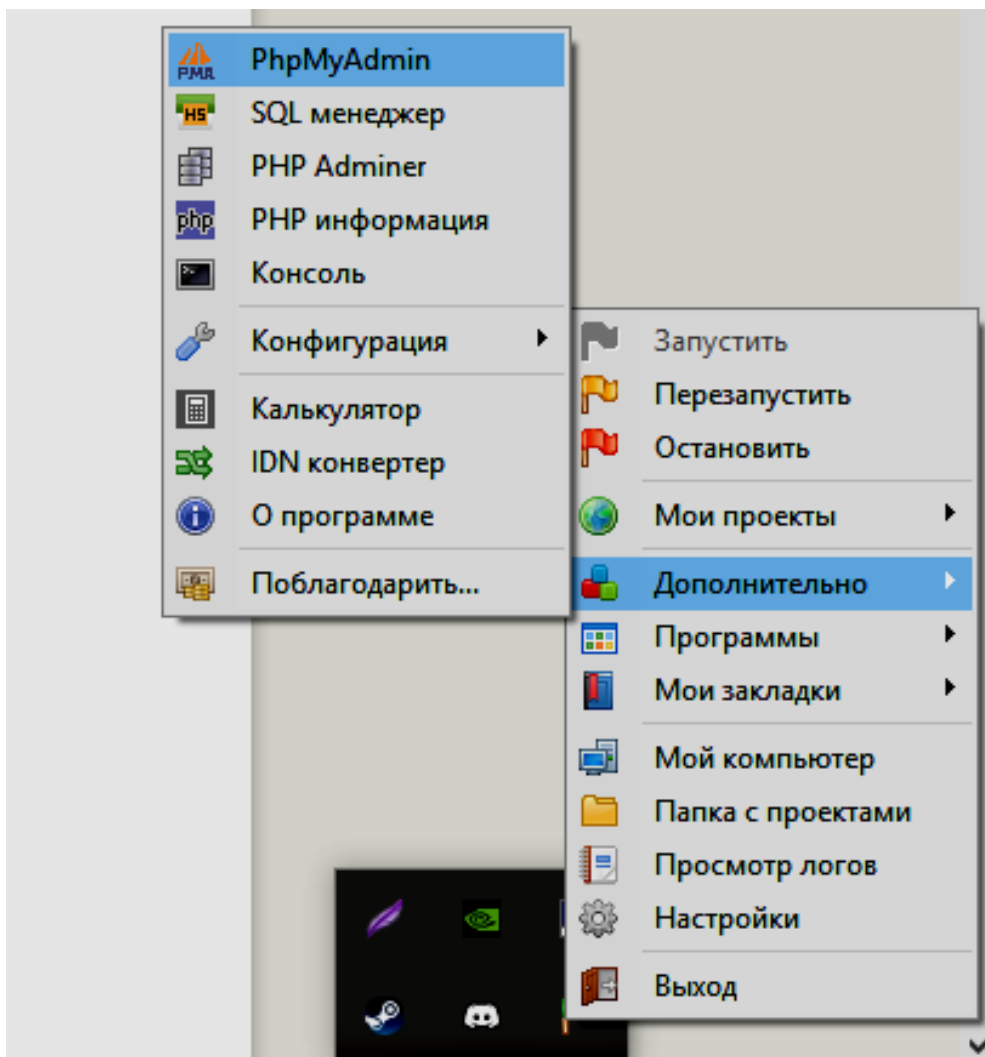


Рисунок 3.14 – Open Server[11]

При встановленні Open Server, обов'язково поставте галочку над пунктом «Встановити модуль «phpMyAdmin» [10], якщо модуль не буде встановлено, потрібно буде перевстановити Open Server.

Після того як у вас запуститься phpMyAdmin, потрібно пройти авторизацію, для цього вкажіть стандартні значення логін «root» і пароль «root», після чого ви попадете на головну сторінку phpMyAdmin яка зображена на рисунку 3.15 .

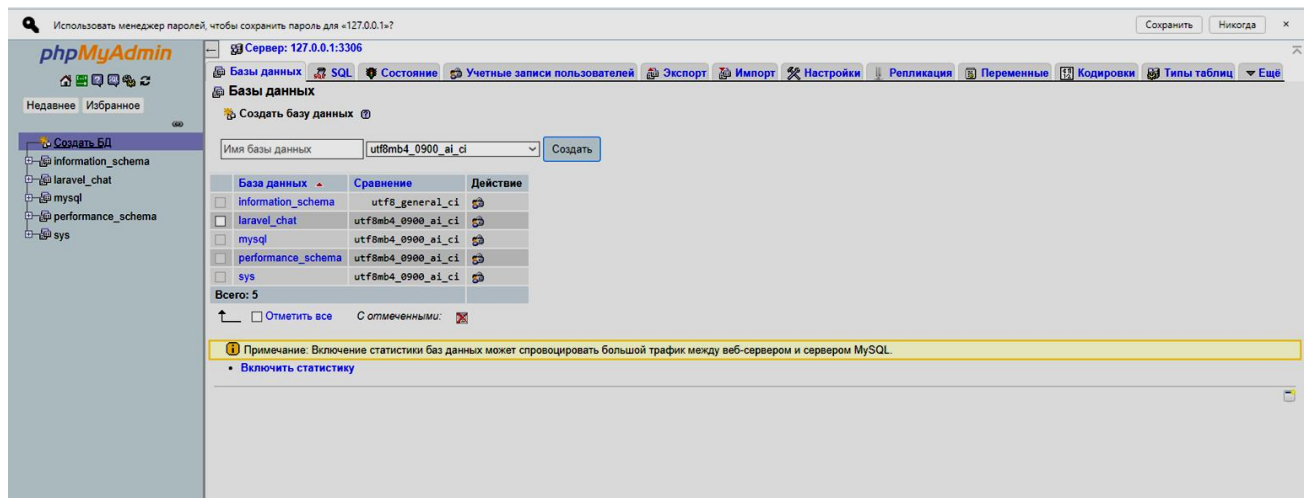


Рисунок 3.15 – Головна сторінка «phpMyAdmin»

На головній сторінці вам будуть відображатися, всі доступні вам бази даних, які колись були вами створені, або добавлені системою за стандартом.

Після чого залишилося створити базу даних для нашої задачі і підключити її до проекту.

Перед початком роботи потрібно перевірити підключення до глобальної мережі інтернет. З'єднання повинно бути стабільно хорошим. Після перевірки наявного підключення до мережі, потрібно відкрити головну сторінку сайту, яка зображена на рисунку 3.16.

Після чого нажимаємо кнопку «Register», і проходимо реєстрацію в системі, вказавши потрібні дані. Форма реєстрації зображена на рисунку 3.17.



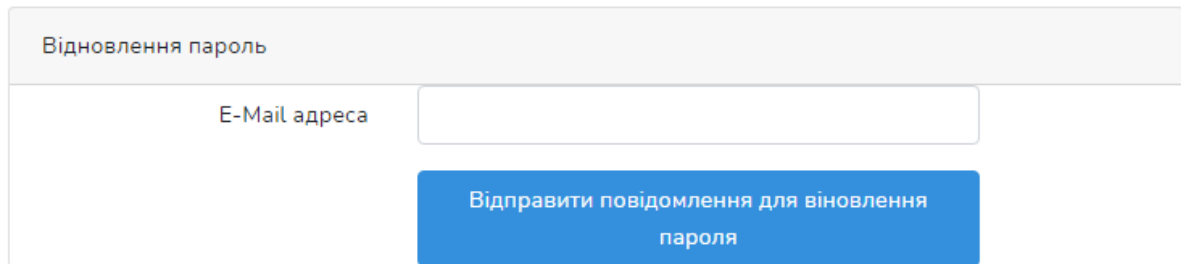
Рисунок 3.16 – Головна сторінка сайту

The image shows a registration form titled 'Register'. It contains four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Each field is a simple rectangular box. Below the fields is a blue button with the text 'Register' in white. The form is set against a light gray background.

Рисунок 3.17 – Форма реєстрації

В ситуації коли користувачу потрібно відновити пароль, він може скористатися цією функцією, яка зображена на рисунку 3.18. В ситуації коли користувача зареєстрований у веб орієнтованій системі по обміну текстовими повідомленнями не може згадати пароль від своєї сторінки, яку він зареєстрував

в системі, тоді він може відновити його, скинувши пароль на свою електронну пошту, яка була вказана при реєстрації користувача в системі.



Відновлення пароль

E-Mail адреса

Відправити повідомлення для віновлення пароля

Рисунок 3.18 – Відновлення пароля користувача

Після успішної реєстрації користувача, він попадає на інтерфейс користувача, де він безпосередньо може вибрати діалог з іншим користувачем, для обміну текстовими повідомленнями в середині системи, яка зображена на рисунку 3.19.

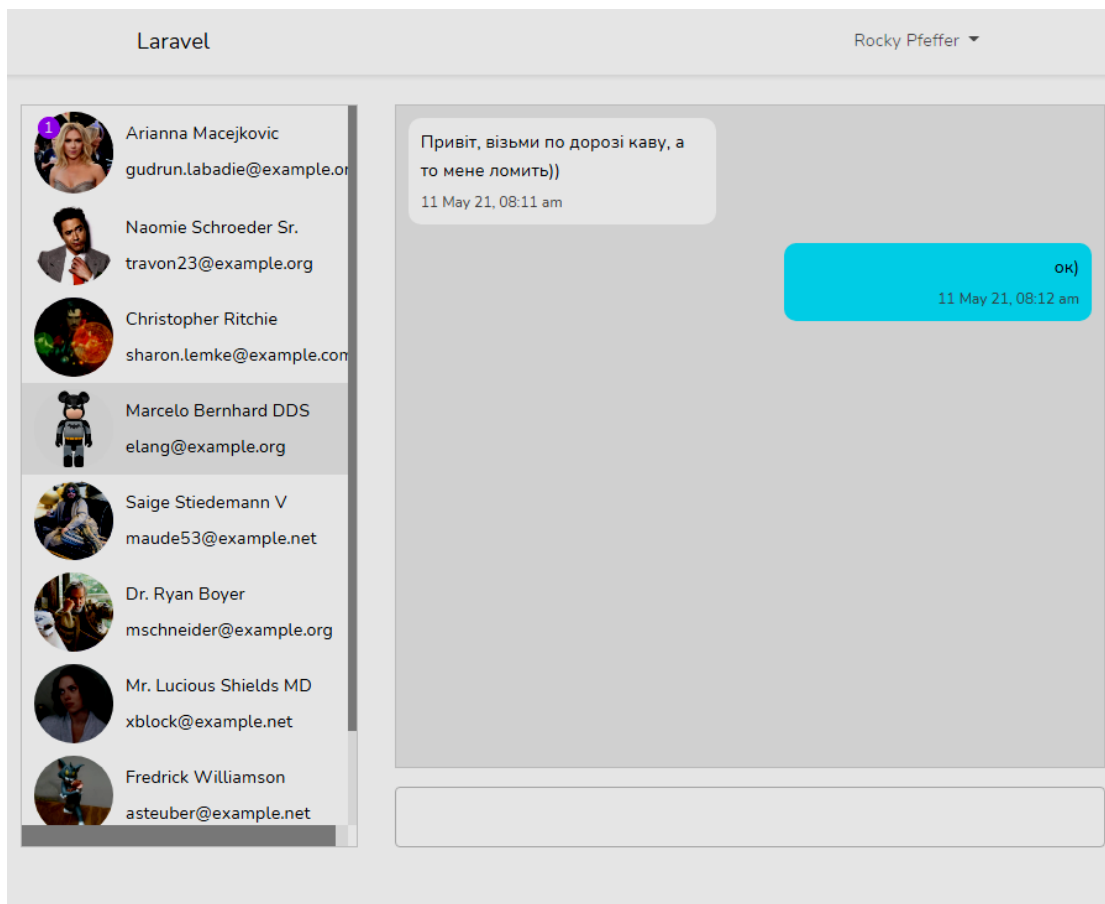


Рисунок 3.19 – Інтерфейс користувача

3.5 Вимоги до розгортання інформаційної системи

Для коректної роботи веб орієнтованого додатку по обміну текстовими повідомленнями, потрібні такі характеристики.

Мінімальні вимоги:

- Операційна система: Microsoft Windows, macOS, Linux;
- Оперативна пам'ять: 512 Mb;
- Розрядність: x86 (32-bit) або x64 (64-bit)
- ЦП: Pentium 4 с SSE2
- Відеоадаптер: 3D адаптер nVidia, Intel, AMD/ATI;
- Відео пам'ять: 64Mb;
- Вінчестер: 350 Mb;
- Контролер: клавіатура і мишка;
- Інтернет: загально доступний;
- Розширення екрану: SVGA 800x600.

Не підтримується Internet Explorer та інші застарілі браузері або старі версії сучасних браузерів.

Сайт не створений для перегляду на мобільних пристроях або інших пристроях з маленьким розміром екрану.

Висновки

В результаті виконання кваліфікаційної роботи бакалавра був отриманий досвід роботи з передовими технологіями розробки веб орієнтованих програмних продуктів. Як наслідок, було спроектовано та програмно реалізовано веб орієнтовну систему обміну текстовими повідомленнями на платформі PHP, що виконує функції реєстрації і авторизації користувачів, роботи з профілем користувача, роботи з відправленням повідомлень, пошук і перегляд відкритої інформації профілів інших користувачів тощо. Веб орієнтовну систему обміну текстовими повідомленнями було створено на мові програмування PHP з використанням програмного каркасу Laravel СКБД MySQL, редактор програмного коду Visual Studio Code.

В кінцевому результаті був одержаний програмний продукт з можливістю обміну текстовими повідомленнями між користувачами, зареєстрованими в системі, з додатковою можливістю збереження всієї доступної інформації між користувачами в БД для подальшого використання. Результат роботи повністю відповідає поставленій задачі. Основна мета проекту, а саме, обмін текстовими повідомленнями відбувається коректно.

В даного веб орієнтовного додатку є великі перспективи в майбутньому, так як потреба в безпечному засобі комунікації між користувачами є актуальною проблемою і на сьогоднішній день. Даний продукт має необмежений спектр модифікації в майбутньому, такі як додавання медіа плеєра для прослуховування аудіо записів, можливість створення Chat-кімнат, анімованої реакції на відповіді, додавання чат-ботів тощо.

Перелік посилань

1. Веб технології. URL: <https://studfile.net/preview/1624161/>
2. Larevel. URL: <https://laravel.ru/>
3. Wikipedia. FrontEnd. URL:
https://uk.wikipedia.org/wiki/Front_end_%D1%82%D0%B0_front_end
4. Wikipedia. BackEnd. URL:
https://uk.wikipedia.org/wiki/Front_end_%D1%82%D0%B0_back_end
5. Що таке форум URL:
http://wiki.fizmat.tnpu.edu.ua/index.php/Що_таке_форум
6. Wikipedia. CSS. URL: <https://uk.wikipedia.org/wiki/CSS>
7. Wikipedia. PHP. URL: <https://uk.wikipedia.org/wiki/PHP>
8. PHP – популярна мова програмування. URL: <https://chili-web.com.ua/php-5/>
9. Мова програмування PHP. URL:
<https://sites.google.com/site/iasaka2516/mova-programuvanna-php>
10. Laravel. URL: <https://cheport.com.ua/blog/cto-takoe-laravel>
11. Google Play. URL: <https://play.google.com>
12. Android. URL: https://www.android.com/intl/ru_ru/
13. App Store. URL: <https://www.apple.com/ua/app-store/>
14. Microsoft Store URL: <https://www.microsoft.com/ru-tj/store/b/home>
15. Whatsapp. URL: <https://www.whatsapp.com/?lang=uk>
16. Telegram. URL: <https://web.telegram.org/#/login>
17. Viber. URL: <https://www.viber.com/ru/>
18. Microsoft .NET. URL: <https://support.microsoft.com/>
19. ASP.NET. URL: asp.net
20. Чому Laravel – найкращий PHP Framework. URL: <https://mcsite.ua/>
21. Wikipedia. FaceBook Messenger. URL:
https://uk.wikipedia.org/wiki/Facebook_Messenger

22. Месенджери або протектори особистого простору. URL: https://kebeta.agency/article/messendzheri_ili__protektori_lichnogo_prostranstva
23. Phpmyadmin. URL: <https://www.phpmyadmin.net>
24. Open Server. URL: <https://ospanel.io/>
25. Історія Телеграму. URL: https://tech.24tv.ua/telegram-bot-shho-tse-istoriya-stvorenniya-oglyad-na-telegram-kanali_n1343625
26. Wikipedia. Чат URL: <https://uk.wikipedia.org/wiki/Чат>
27. Wikipedia. HTML. URL: <https://uk.wikipedia.org/wiki/HTML>
28. Wikipedia. Соціальні мережі URL: [https://uk.wikipedia.org/wiki/Соціальна_мережа_\(Інтернет\)](https://uk.wikipedia.org/wiki/Соціальна_мережа_(Інтернет))
29. MySQL. URL: <https://www.mysql.com/>

ДОДАТКИ

Додаток А

Програмні коди

HomeController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Message;
```

```
use App\User;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Facades\Auth;
```

```
use Illuminate\Support\Facades\DB;
```

```
use Pusher\Pusher;
```

```
class HomeController extends Controller
```

```
{
```

```
    /**
```

```
     *
```

```
     * @return void
```

```
     */
```

```
    public function __construct()
```

```
    {
```

```
        $this->middleware('auth');
```

```
    }
```

```
    *
```

```
    * @return \Illuminate\Contracts\Support\Renderable
```

```
    */
```

```
    public function index()
```

```
{
```

```
        $users = DB::select("select users.id, users.name, users.avatar, users.email, count(is_read) as unread  
        from users LEFT JOIN messages ON users.id = messages.from and is_read = 0 and messages.to = " . Auth::id() . "  
        where users.id != " . Auth::id() . "  
        group by users.id, users.name, users.avatar, users.email");
```

```
        return view('home', ['users' => $users]);
```

```

}

public function getMessage($user_id)
{
    $my_id = Auth::id();

    Message::where(['from' => $user_id, 'to' => $my_id])->update(['is_read' => 1]);

    $messages = Message::where(function ($query) use ($user_id, $my_id) {
        $query->where('from', $user_id)->where('to', $my_id);
    }->orWhere(function ($query) use ($user_id, $my_id) {
        $query->where('from', $my_id)->where('to', $user_id);
    }->get());

    return view('messages.index', ['messages' => $messages]);
}

public function sendMessage(Request $request)
{
    $from = Auth::id();
    $to = $request->receiver_id;
    $message = $request->message;

    $data = new Message();
    $data->from = $from;
    $data->to = $to;
    $data->message = $message;
    $data->is_read = 0;    $data->save();

    // pusher
    $options = array(
        'cluster' => 'ap2',
        'useTLS' => true
    );

    $pusher = new Pusher(
        env('PUSHER_APP_KEY'),
        env('PUSHER_APP_SECRET'),
        env('PUSHER_APP_ID'),
        $options
    );
}

```

```

        $data = ['from' => $from, 'to' => $to];    $pusher->trigger('my-channel', 'my-event', $data);
    }
}

```

ContactsController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\User;
```

```
use App\Message;
```

```
use App\Events\NewMessage;
```

```
class ContactsController extends Controller
```

```
{
```

```
    public function get()
```

```
        $contacts = User::where('id', '!=', auth()->id()->get());
```

```
        $unreadIds = Message::select(\DB::raw("`from` as sender_id, count(`from`) as messages_count"))
```

```
            ->where('to', auth()->id())
```

```
            ->where('read', false)
```

```
            ->groupBy('from')
```

```
            ->get();
```

```
        $contacts = $contacts->map(function($contact) use ($unreadIds) {
```

```
            $contactUnread = $unreadIds->where('sender_id', $contact->id)->first();
```

```
            $contact->unread = $contactUnread ? $contactUnread->messages_count : 0;
```

```
            return $contact;
```

```
        });
```

```
        return response()->json($contacts);
```

```
    }
```

```
    public function getMessagesFor($id)
```

```
    {
```

```
        Message::where('from', $id)->where('to', auth()->id()->get()->id())->update(['read' => true]);
```

```
        $messages = Message::where(function($q) use ($id) {
```

```
            $q->where('from', auth()->id());
```

```

        $q->where('to', $id);
    }->orWhere(function($q) use ($id) {
        $q->where('from', $id);
        $q->where('to', auth()->id());
    })
    ->get();

    return response()->json($messages);
}

public function send(Request $request)
{
    $message = Message::create([
        'from' => auth()->id(),
        'to' => $request->contact_id,
        'text' => $request->text
    ]);

    broadcast(new NewMessage($message));

    return response()->json($message);
}
}

```

Kernel.php

```
<?php
```

```
namespace App\Http;
```

```
use Illuminate\Foundation\Http\Kernel as HttpKernel;
```

```
class Kernel extends HttpKernel
```

```
{
```

```
/**
```

```
*
```

```
* @var array
```

```
*/
```

```
protected $middleware = [
```

```
    \App\Http\Middleware\TrustProxies::class,
```

```
    \App\Http\Middleware\CheckForMaintenanceMode::class,
```

```
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
```

```

\App\Http\Middleware\TrimStrings::class,
\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
];

*
* @var array
*/
protected $middlewareGroups = [
    'web' => [
        \App\Http\Middleware\EncryptCookies::class,
        \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
        \Illuminate\Session\Middleware\StartSession::class,
        // \Illuminate\Session\Middleware\AuthenticateSession::class,
        \Illuminate\View\Middleware\ShareErrorsFromSession::class,
        \App\Http\Middleware\VerifyCsrfToken::class,
        \Illuminate\Routing\Middleware\SubstituteBindings::class,
    ],

    'api' => [
        'throttle:60,1',
        'bindings',
    ],
];

*
* @var array
*/
protected $routeMiddleware = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];

protected $middlewarePriority = [
    \Illuminate\Session\Middleware\StartSession::class,

```

```

    \Illuminate\View\Middleware\ShareErrorsFromSession::class,
    \App\Http\Middleware\Authenticate::class,
    \Illuminate\Routing\Middleware\ThrottleRequests::class,
    \Illuminate\Session\Middleware\AuthenticateSession::class,
    \Illuminate\Routing\Middleware\SubstituteBindings::class,
    \Illuminate\Auth\Middleware\Authorize::class,
];
}

```

User.php

```

<?php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use Notifiable;

    /**
     *
     * @var array
     */
    protected $fillable = [
        'name', 'avatar', 'email', 'password',
    ];

    /**
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**

```

```

*
* @var array
*/
protected $casts = [
    'email_verified_at' => 'datetime',
];
}

```

AuthServiceProvider.php

```
<?php
```

```

namespace Illuminate\Auth;

use Illuminate\Auth\Access\Gate;
use Illuminate\Contracts\Auth\Access\Gate as GateContract;
use Illuminate\Contracts\Auth\Authenticatable as AuthenticatableContract;
use Illuminate\Support\ServiceProvider;

class AuthServiceProvider extends ServiceProvider
{
    /**
     * Register the service provider.
     *
     * @return void
     */
    public function register()
    {
        $this->registerAuthenticator();
        $this->registerUserResolver();
        $this->registerAccessGate();
        $this->registerRequestRebindHandler();
        $this->registerEventRebindHandler();
    }

    /**
     * Register the authenticator services.
     *
     * @return void
     */
    protected function registerAuthenticator()
    {
        $this->app->singleton('auth', function ($app) {
            $app['auth.loaded'] = true;

```

```

        return new AuthManager($app);
    });

    $this->app->singleton('auth.driver', function ($app) {
        return $app['auth']->guard();
    });
}

/**
 *
 * @return void
 */
protected function registerUserResolver()
{
    $this->app->bind(
        AuthenticatableContract::class, function ($app) {
            return call_user_func($app['auth']->userResolver());
        }
    );
}

/**
 *
 * @return void
 */
protected function registerAccessGate()
{
    $this->app->singleton(GateContract::class, function ($app) {
        return new Gate($app, function () use ($app) {
            return call_user_func($app['auth']->userResolver());
        });
    });
}

/**
 *
 * @return void
 */
protected function registerRequestRebindHandler()
{
    $this->app->rebinding('request', function ($app, $request) {
        $request->setUserResolver(function ($guard = null) use ($app) {

```

```

        return call_user_func($app['auth']->userResolver(), $guard);
    });
});
}

/**
 *
 * @return void
 */
protected function registerEventRebindHandler()
{
    $this->app->rebinding('events', function ($app, $dispatcher) {
        if (!$app->resolved('auth')) {
            return;
        }

        if ($app['auth']->hasResolvedGuards() === false) {
            return;
        }

        if (method_exists($guard = $app['auth']->guard(), 'setDispatcher')) {
            $guard->setDispatcher($dispatcher);
        }
    });
}
}

```

CreatesUserProviders.php

```

<?php

namespace Illuminate\Auth;

use InvalidArgumentException;

trait CreatesUserProviders
{
    /**
     * The registered custom provider creators.
     *
     * @var array
     */
    protected $customProviderCreators = [];
}

```

```

/**

*
* @param string|null $provider
* @return \Illuminate\Contracts\Auth\UserProvider|null
*
* @throws \InvalidArgumentException
*/
public function createUserProvider($provider = null)
{
    if (is_null($config = $this->getProviderConfiguration($provider))) {
        return;
    }

    if (isset($this->customProviderCreators[$driver = ($config['driver'] ?? null)])) {
        return call_user_func(
            $this->customProviderCreators[$driver], $this->app, $config
        );
    }

    switch ($driver) {
        case 'database':
            return $this->createDatabaseProvider($config);
        case 'eloquent':
            return $this->createEloquentProvider($config);
        default:
            throw new InvalidArgumentException(
                "Authentication user provider [{$driver}] is not defined."
            );
    }
}

/**
* Get the user provider configuration.
*
* @param string|null $provider
* @return array|null
*/
protected function getProviderConfiguration($provider)
{
    if ($provider = $provider ?: $this->getDefaultUserProvider()) {
        return $this->app['config']['auth.providers.'.$provider];
    }
}

```

```

}

/**

*
* @param array $config
* @return \Illuminate\Auth\DatabaseUserProvider
*/
protected function createDatabaseProvider($config)
{
    $connection = $this->app['db']->connection($config['connection'] ?? null);

    return new DatabaseUserProvider($connection, $this->app['hash'], $config['table']);
}

/**

*
* @param array $config
* @return \Illuminate\Auth\EloquentUserProvider
*/
protected function createEloquentProvider($config)
{
    return new EloquentUserProvider($this->app['hash'], $config['model']);
}

/**
 * Get the default user provider name.
 *
 * @return string
 */
public function getDefaultUserProvider()
{
    return $this->app['config']['auth.defaults.provider'];
}
}

```

MustVerifyEmail.php

```
<?php
```

```
namespace Illuminate\Auth;
```

```
trait MustVerifyEmail
```

```
{  
  /**  
   * Determine if the user has verified their email address.  
   *  
   * @return bool  
   */  
  public function hasVerifiedEmail()  
  {  
    return !is_null($this->email_verified_at);  
  }  
  
  /**  
   * Mark the given user's email as verified.  
   *  
   * @return bool  
   */  
  public function markEmailAsVerified()  
  {  
    return $this->forceFill([  
      'email_verified_at' => $this->freshTimestamp(),  
    ]->save());  
  }  
  
  /**  
   *  
   * @return void  
   */  
  public function sendEmailVerificationNotification()  
  {  
    $this->notify(new Notifications\VerifyEmail);  
  }  
  
  /**  
   *  
   * @return string  
   */  
  public function getEmailForVerification()  
  {  
    return $this->email;  
  }  
}
```

Recaller.php

```
<?php

namespace Illuminate\Auth;

use Illuminate\Support\Str;

class Recaller
{
    /**
     * The "recaller" / "remember me" cookie string.
     *
     * @var string
     */
    protected $recaller;

    /**
     *
     * @param string $recaller
     * @return void
     */
    public function __construct($recaller)
    {
        $this->recaller = @unserialize($recaller, ['allowed_classes' => false]) ?: $recaller;
    }

    /**
     *
     * @return string
     */
    public function id()
    {
        return explode('|', $this->recaller, 3)[0];
    }

    /**
     *
     * @return string
     */
}
```

```

public function token()
{
    return explode('|', $this->recaller, 3)[1];
}
*
* @return string
*/
public function hash()
{
    return explode('|', $this->recaller, 3)[2];
}
*
* @return bool
*/
public function valid()
{
    return $this->properString() && $this->hasAllSegments();
}

*
* @return bool
*/
protected function properString()
{
    return is_string($this->recaller) && Str::contains($this->recaller, '|');
}
* Determine if the recaller has all segments.
*
* @return bool
*/
protected function hasAllSegments()
{
    $segments = explode('|', $this->recaller);

    return count($segments) === 3 && trim($segments[0]) !== "" && trim($segments[1]) !== "";
}
}

```

Додаток Б

Презентаційний матеріал

Кваліфікаційна робота бакалавра

“Веб орієнтовна система обміну текстовими повідомленнями”

Виконав: студент групи КН-17-2 Козакевич В. А.
Керівник : к.т.н., доц. кафедри КНІТ Мазурець О. В.

АКТУАЛЬНІСТЬ ТЕМИ

Використання месенджера дозволить автоматизувати багато процесів, що дозволить пришвидшити та спростити роботу користувача в багатьох галузях роботи. Месенджер дозволить з легкістю назначати зустрічі, проводити конференції, та підтримувати зв'язок між користувачами.

Месенджер — телекомунікаційна служба для обміну текстовими повідомленнями між комп'ютерами або іншими пристроями користувачів через комп'ютерні мережі

ЗАВДАННЯ ПРОЕКТУ

Метою даного проекту є розробка соціальної мережі на платформі PHP, що виконує наступні основні функції:

- ✓ реєстрація користувачів та робота з ними (реєстрація і авторизація користувачів);
- ✓ робота з особистими даними (редагування даних профілю);
- ✓ робота з повідомленнями (відправлення, одержання, формування діалогів, переліку діалогів)
- ✓ робота з профілями інших користувачів (пошук, перегляд відкритої інформації).

UML-ДІАГРАМА ПРОЕКТУ

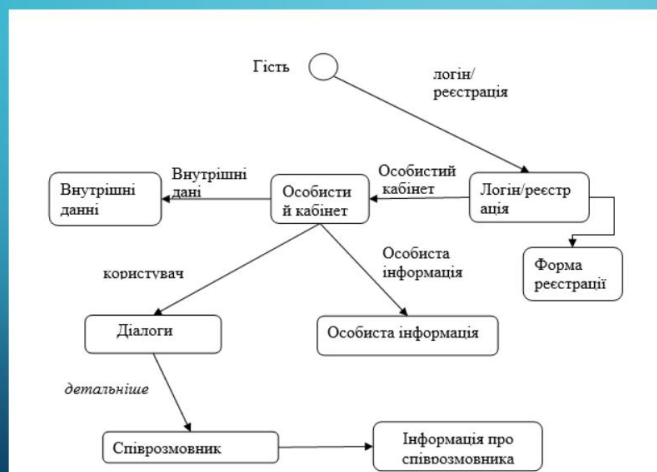
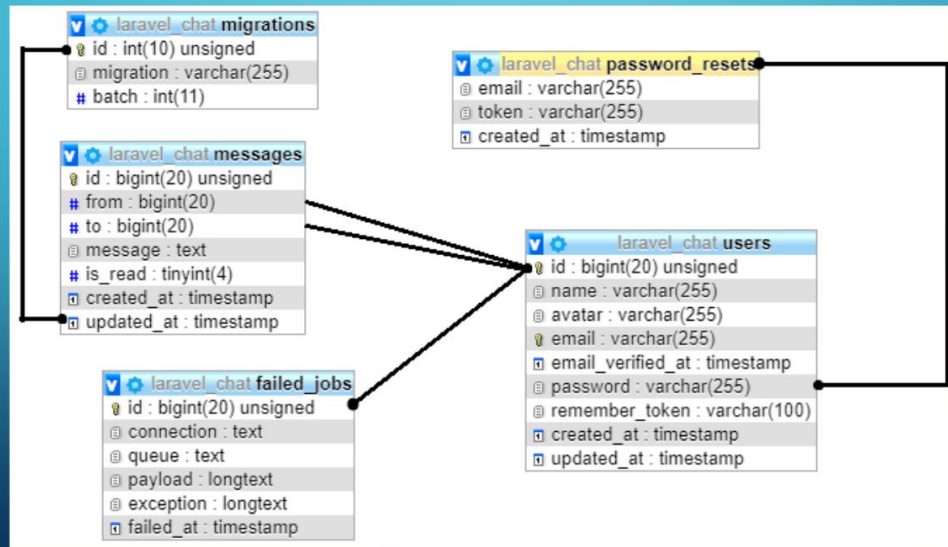
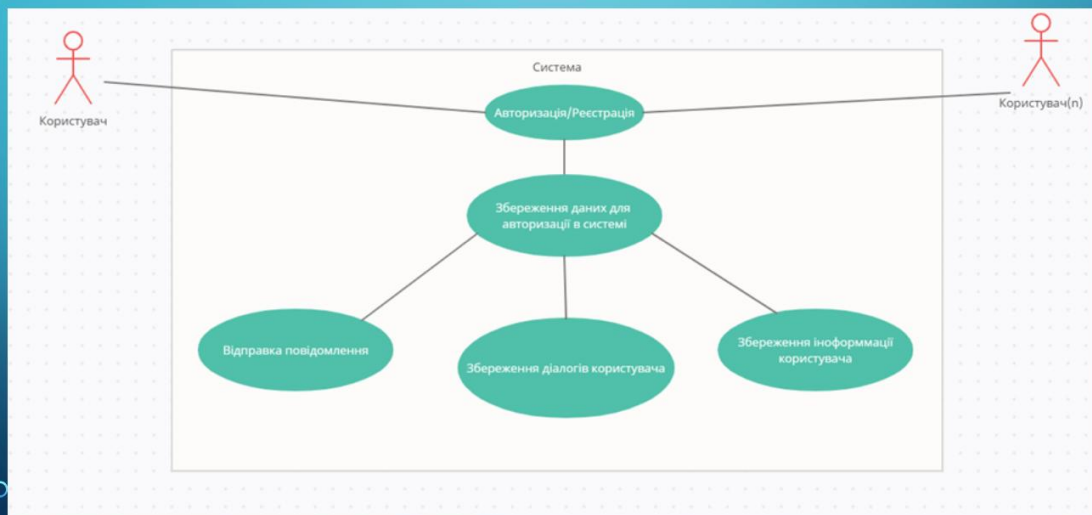


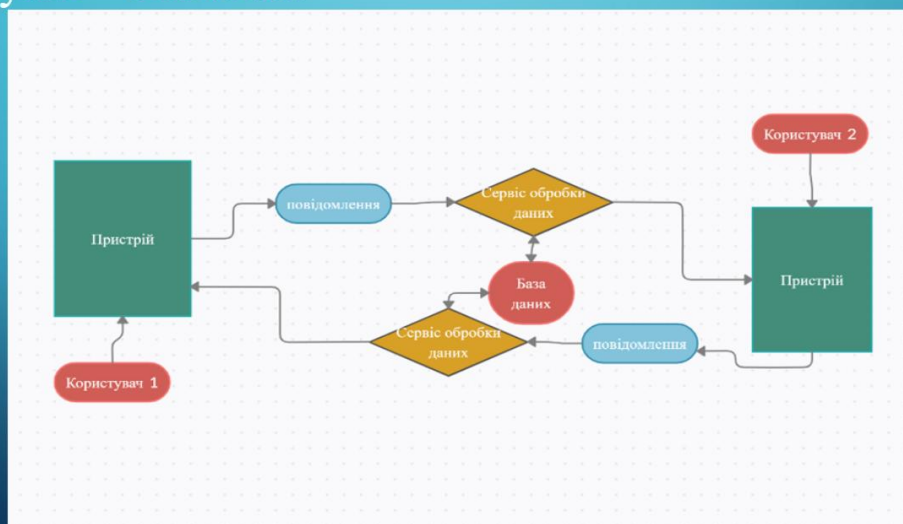
Схема бази даних



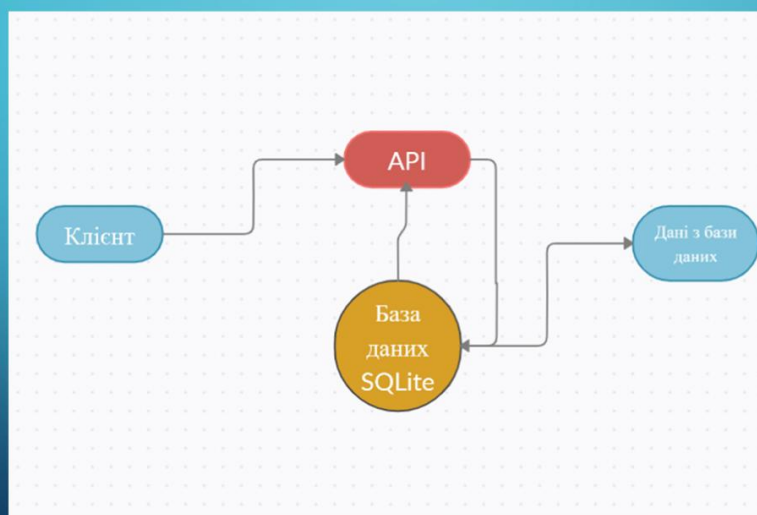
UML-діаграма бізнес-процесів сайту



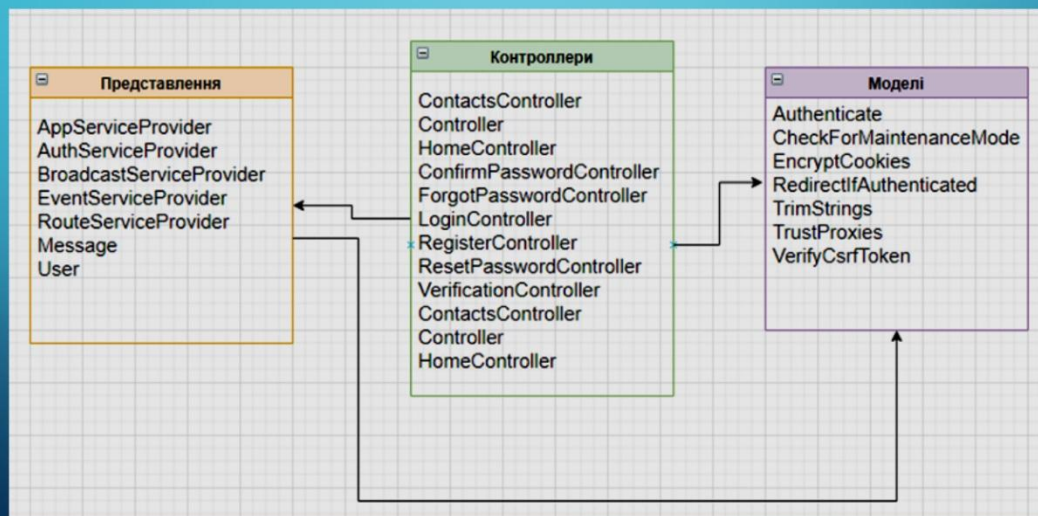
Структура і функціональне призначення модулів системи



Структура функціонування системи API



Діаграма класів



ПЛАТФОРМА

Даний проект використовує платформу PHP.

PHP (англ. PHP: Hypertext Preprocessor — PHP: гіпертекстовий препроцесор), попередня назва: *Personal Home Page Tools* — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта.



БАЗА ДАНИХ

Проект використовує MySQL в якості бази даних.

MySQL — вільна система керування реляційними базами даних.

Це — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.



ФРЕЙМВОРК

Даний проект використовує фреймворк Laravel.

Laravel — безкоштовний, з відкритим кодом PHP-фреймворк, створений Taylor Otwell і призначений для розробки веб-додатків відповідно до шаблону model-view-controller (MVC). Деякі з особливостей Laravel є модульна система упакування з виділеним менеджером залежностей Composer, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні додатків і технічного обслуговування.

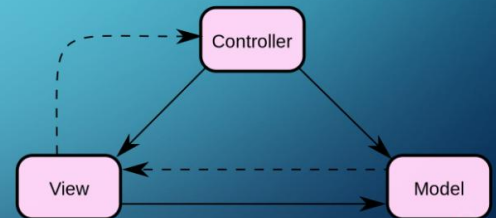


АРХІТЕКТУРА ПРОЕКТУ

Даний проект використовує архітектуру MVC або Model View Controller.

Суть цієї архітектури полягає в розділенні програми на моделі, контролери та представлення.

У Моделі реалізується бізнес-логіка програми. У представленні знаходиться вивід даних системи. Контролер реалізує зв'язок між моделями та представленнями.



Аутентифікація користувача

На даному слайді продемонстрована авторизація користувача в системі

Laravel Login Register

Login

E-Mail Address

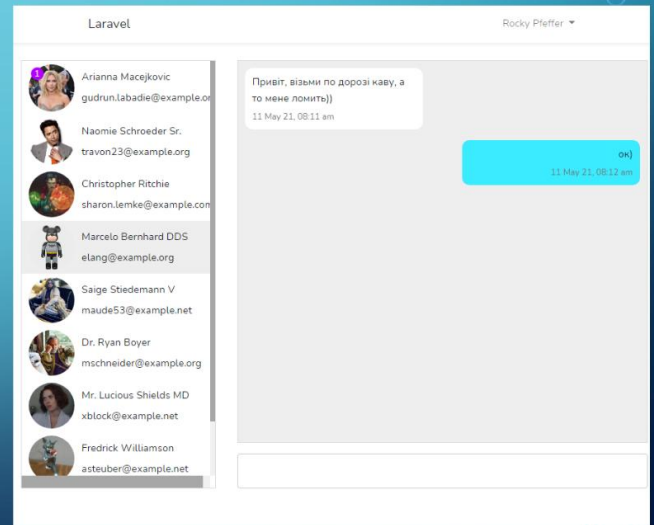
Password

Remember Me

[Forgot Your Password?](#)

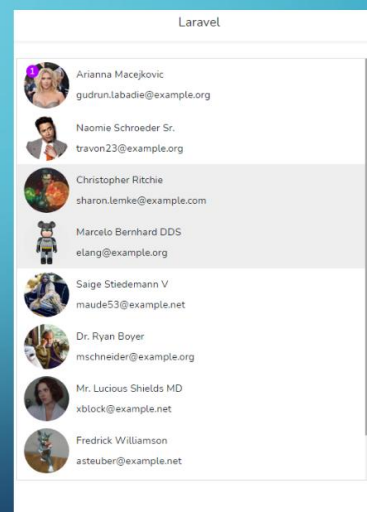
Головна сторінка користувача

Інтерфейс головної сторінки користувача



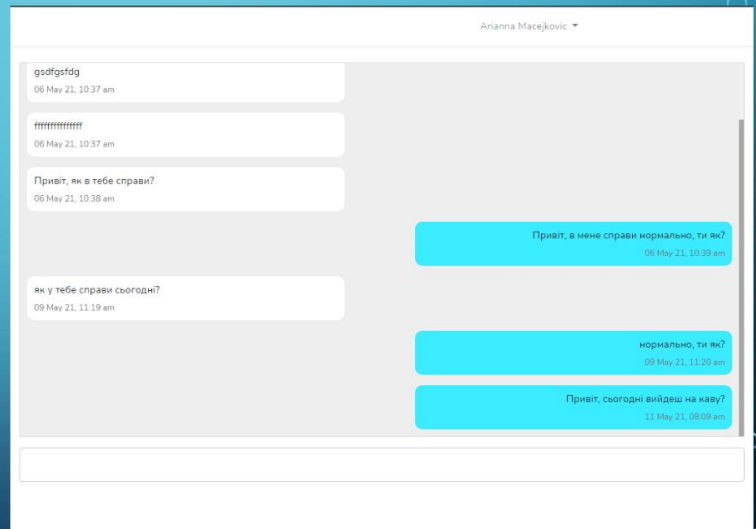
Список контактів

Список користувачів, доступних для взаємного обміну повідомленнями



Діалогове вікно

На слайді зображий діалог між двома зареєстрованими користувачами системи, із зображенням імя користувача та числом і датою відправленого повідомлення



ВИСНОВКИ

- В результаті виконання кваліфікаційної роботи бакалавра був отриманий досвід роботи з передовими технологіями розробки веб орієнтованих програмних продуктів. Як наслідок, було спроектовано та програмно реалізовано веб орієнтовану систему обміну текстовими повідомленнями на платформі PHP, що виконує функції реєстрації і авторизації користувачів, роботи з профілем користувача, роботи з відправленням повідомлень, пошук і перегляд відкритої інформації профілів інших користувачів тощо. Веб орієнтовану систему обміну текстовими повідомленнями було створено на мові програмування PHP з використанням програмного каркасу Laravel СКБД MySQL, редактор програмного коду Visual Studio Code.
- В кінцевому результаті був одержаний програмний продукт з можливістю обміну текстовими повідомленнями між користувачами, зареєстрованими в системі, з додатковою можливістю збереження всієї доступної інформації між користувачами в БД для подальшого використання. Результат роботи повністю відповідає поставленій задачі. Основна мета проекту, а саме, обмін текстовими повідомленнями відбувається коректно.
- В даного веб орієнтованого додатку є великі перспективи в майбутньому, так як потреба в безпечному засобі комунікації між користувачами є актуальною проблемою і на сьогоднішній день. Даний продукт має необмежений спектр модифікації в майбутньому, такі як додавання медіа плеєра для прослуховування аудіо записів, можливість створення Chat-кімнат, анімованої реакції на відповіді, додавання чат-ботів тощо.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Козакевич В. А. на захист дипломного проекту (роботи)
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Веб орієнтовна система обміну текстовими повідомленнями

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О.С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Козакевич В. А. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:
національною шкалою: відмінно 18,75 %, добре 68,75 %, задовільно 12,50 %.
шкалою ЄКТС: А 12,73 %, В 16,36 %, С 50,91 %, D 10,91 %, E 9,09 %.

Методист факультету

[Signature]

(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Козакевич В. А. виконав комп'ютерний проект
за актуальною тематикою, використавши сучасні
інструментарій програмування та середовища
розробки. Всі вимоги в роботі задані повністю.
Пра виконанні роботи Козакевич В. А. вчасно і
на високому професійному рівні виконав
всі необхідні етапи.

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

[Signature]

(підпис)

Мартинчук О.В.

(прізвище та ініціали)

" 08 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Козакевич В. А. допускається до захисту цього

Завідувач кафедри

КНІТ

(назва)

" 08 " 06 2021 р.

(підпис, прізвище, ініціали)

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 36.0%

Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 11%

ID: 92901 Название: Веб орієнтовна система обміну текстовими повідомленнями Добавлено в БД: 2021-06-09 Авторы: В.А. Козакевич Руководители: О.В. Мазурець Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	59624	521	22516 (38%)	215 (41%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы
90159	Название: ЗВІТ з професійної практики Добавлено в БД: 2021-05-11 Авторы: Козакевич В.А. Руководители: Скрипник Т.К. Консультанты: Оponentы:	21334 (36.0%)	206 (40.0%)

Ім'я користувача:
Кафедра КН

Дата перевірки:
09.06.2021 15:24:52 EEST

Дата звіту:
10.06.2021 12:51:38 EEST

ID перевірки:
1008246414

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005671

Назва документа: 2021 КРБ Козакевич 20210607 7 фінал Lite

Кількість сторінок: 56 Кількість слів: 8709 Кількість символів: 64448 Розмір файлу: 2.01 MB ID файлу: 1008318642

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

7.18% Схожість

Найбільша схожість: 3.35% з джерелом з Бібліотеки (ID файлу: 1008318434)

2.54% Джерела з Інтернету

89

Сторінка 58

5.11% Джерела з Бібліотеки

54

Сторінка 58

0% Цитат

Вилучення цитат виключено

Вилучення списку бібліографічних посилань виключено

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

3

Підозріле форматування

12
сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Веб орієнтовна система обміну текстовими повідомленнями

Автор: студент групи КН-17-2 Козакевич Віталій Анатолійович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: доцент кафедри КНІТ Мазурець О.В.

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.	<i>відповідає</i>
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 безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що не мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 7,18 % і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

О. В. Мазурець

Гарант ОП

О. В. Мазурець

Завідувач кафедри КНІТ

О. В. Бармак

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-17-2 Козакевича Віталія Анатолійовича

за темою Веб орієнтовна система обміну текстовими повідомленнями

1. Актуальність і значення теми На сьогоднішній день існує велика кількість різновидів веб орієнтованих додатків по обміну інформацією. Всі вони відрізняються в певній мірі функціоналом, але залишають за собою одну спільну мету, це обмін інформацією між користувачами. Такі додатки широко використовуються у всіх галузях виробництва та в спілкуванні, тому даний напрям інформаційних технологій є безсумнівно перспективним.

2. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки Створена в роботі функціональна структура інформаційної системи та наведені шляхи автоматизації бізнес-процесів предметної області забезпечують можливість реєстрації користувачів, відправлення-одержання повідомлень та формування діалогів користувачами системи, що є основними функціями веб орієнтовних систем обміну текстовими повідомленнями. При розробці інформаційної системи використано сучасні й ефективні засоби – мову програмування PHP з використанням програмного каркасу Laravel, СКБД MySQL і редактор програмного коду Visual Studio Code.

3. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність Розроблена інформаційна система дозволяє виконувати функції реєстрації і авторизації користувачів, роботи з профілем користувача, роботи з відправленням повідомлень, виконувати пошук і перегляд відкритої інформації профілів інших користувачів тощо. Результат роботи повністю відповідає поставленій задачі.

4. Загальний висновок та оцінка Всі поставлені завдання виконані повністю і на високому професійному рівні. За своєю структурою, практичними цінностями, поставленій меті та вирішеними завданнями робота відповідає вимогам вищої школи і вимогам, що пред'являються до освітньо-кваліфікаційного рівня «бакалавр», а її автор Шичко А.В. заслуговує присвоєння кваліфікації бакалавра з комп'ютерних наук.

Робота заслуговує на оцінку « добре ».

Рецензент д.т.н., доц. д.с.ф. ТМІТ Шичко А.В.