

УДК 04.092

DOI: 10.31673/2786-8362.2025.017599

Шикула О.М., д.ф.-м.н.; Іщеряков С.М., к.т.н.;
Савончук Д.В., Антонов В.В.

ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ПРИ РОЗРОБЦІ СИСТЕМИ МИТТЄВОЇ ПІДТРИМКИ КОРИСТУВАЧІВ НА ОСНОВІ PHP ТА AJAX

Shykula O.M., Ishcheryakov S.M., Savonchuk D.V., Antonov V.V. Using cloud technologies in the development of an instant user support system based on PHP and Ajax. The article reviews and analyzes existing instant support systems, analyzes user needs and requirements for instant support. As a result of the analysis, key functions and capabilities that should be available in the system were identified.

An instant support system based on the PHP programming language and the use of Ajax technology to ensure speed and interactivity was designed and developed. The use of these technologies allowed for high-speed processing of requests and providing answers without the need to reload the page, which significantly increases the convenience of using the system.

The developed instant support system has great potential for use in various industries where it is necessary to provide fast and high-quality user service. The possibilities for further development and improvement of the system, in particular through integration with advanced artificial intelligence algorithms and the development of an adaptive interface, open up new prospects for improving the quality of service and meeting user needs.

Keywords: instant support system, requirements analysis, system development, PHP, JavaScript, Ajax, SQLite DBMS

Шикула О.М., Іщеряков С.М., Савончук Д.В., Антонов В.В. Використання хмарних технологій при розробці системи миттєвої підтримки користувачів на основі PHP та Ajax. В статті було проведено огляд і аналіз існуючих систем миттєвої підтримки, аналіз потреб та вимог користувачів щодо миттєвої підтримки. В результаті аналізу були визначені ключові функції та можливості, які мали б бути доступні в системі.

Спроектowana та розроблена система миттєвої підтримки, яка базується на мові програмування PHP та використанні технології Ajax для забезпечення швидкості та інтерактивності. Використання цих технологій дозволило забезпечити високу швидкість обробки запитів та надання відповідей без потреби у перезавантаженні сторінки, що суттєво підвищує зручність користування системою.

Розроблена система миттєвої підтримки має великий потенціал для використання у різних галузях, де необхідно забезпечити швидке та якісне обслуговування користувачів. Можливості подальшого розвитку та удосконалення системи, зокрема через інтеграцію з розширеними алгоритмами штучного інтелекту та розробку адаптивного інтерфейсу, відкривають нові перспективи для покращення якості обслуговування та задоволення потреб користувачів.

Ключові слова: система миттєвої підтримки, аналіз вимог, розробка системи, PHP, JavaScript, Ajax, СУБД SQLite

Вступ

З розвитком інтернет-технологій та зростанням конкуренції сучасні компанії акцентують увагу на забезпеченні високоякісної та оперативної підтримки своїх клієнтів в онлайн-середовищі. В цьому контексті стає надзвичайно важливим та актуальним створення ефективних механізмів миттєвої підтримки, які спроможні негайно реагувати на запити користувачів та забезпечувати вирішення їхніх проблем [1].

При розробці системи миттєвої підтримки користувачів доцільно використовувати сучасні веб-технології. Ключ до успіху сучасної системи миттєвої підтримки полягає у збалансуванні між потужними функціональними можливостями та простотою використання. Також потрібно звернути увагу на вартість інструментів та їхню доступність для різних сегментів ринку.

Створення ефективної системи миттєвої підтримки користувачів, яка буде легко інтегрувати і масштабувати в залежності від потреб бізнесу, буде корисно для компаній, які надають онлайн-сервіси та продукти, оскільки ефективна система миттєвої підтримки дозволить підвищити задоволеність користувачів та зменшити час вирішення їхніх проблем [2].

Постановка завдання. Основними задачами цього дослідження є:

- проведення огляду і аналізу існуючих систем миттєвої підтримки, аналізу потреб та вимог користувачів щодо миттєвої підтримки; дослідження функціоналу, що найбільше цікавить користувачів та спрощує їхню взаємодію з системою, та визначення на цій основі ключових функцій та можливостей, які мали б бути доступні в системі;
- обґрунтування вибору технологій для реалізації проекту: мови програмування PHP та технології Ajax для реалізації проекту;
- проектування, розробка та впровадження системи миттєвої підтримки користувачів на основі вищезгаданих технологій.

Аналіз останніх досліджень. У процесі розроблення системи миттєвої підтримки важливим етапом є аналіз існуючих рішень на ринку. Це дозволить ідентифікувати сильні та слабкі сторони конкурентів та виявити потенційні можливості для інновацій. Огляд кількох популярних систем підтримки користувачів надасть змогу глибше зрозуміти вимоги до сучасних сервісів обслуговування клієнтів та визначити ключові напрямки для розвитку власної системи.

LiveChat – це одна з відомих платформ для надання живої підтримки, яка інтегрується з різними вебсайтами та дозволяє операторам ефективно комунікувати з користувачами у реальному часі. Її переваги полягають у швидкості відгуку, високому рівні налаштування інтерфейсу та можливості інтеграції з іншими сервісами, такими як CRM-системи. Однак, сервіс може бути досить дорогим для малого та середнього бізнесу, і не завжди гарантує повноцінне рішення для складних задач обслуговування.

Zendesk Chat – ще один популярний інструмент, який забезпечує широкий спектр функціональності для комунікації з користувачами. Особливостями Zendesk Chat є її гнучкість і масштабованість, яка дозволяє використовувати її як для невеликих так і для великих компаній. Також, сервіс надає розширену аналітику та звітність, що допомагає краще розуміти потреби клієнтів. Тим не менше, інтерфейс може здатися складним для нових користувачів, а ціна – високою для стартапів.

Intercom – інноваційна платформа, яка об'єднує в собі елементи живого чату, маркетингу та служби підтримки. Її унікальною особливістю є використання штучного інтелекту для автоматизації відповідей та сегментації користувачів, що дозволяє надавати більш персоналізований досвід. Однак, Intercom також може виявитися занадто складним у налаштуванні та вимагати значних інвестицій для повного розгортання та інтеграції з вже існуючими системами підприємства.

Drift – це платформа для залучення відвідувачів сайту, яка використовує "розумні" чат-боти, що можуть ініціювати розмову з потенційними клієнтами. Це допомагає збільшити конверсію та поліпшити досвід користувачів завдяки швидкому та ефективному спілкуванню. Drift пропонує також широкі можливості для персоналізації діалогів залежно від поведінки користувачів на сайті. Основними недоліками можуть бути висока вартість підписки та складність налаштування деталізованих сценаріїв для чат-ботів.

Tidio Chat – інструмент, який дозволяє інтегрувати на сайті функцію живого чату з можливістю додавання чат-ботів. Завдяки гнучким налаштуванням і простоті інтеграції, Tidio став популярним серед малих та середніх підприємств. Він також пропонує базовий безкоштовний план, що робить його доступним для стартапів. Однак, розширені функції вимагають платної підписки, і можливості штучного інтелекту та автоматизації можуть бути обмежені в порівнянні з більш дорогими платформами.

На основі аналізу аналогів можна зробити висновок, що системи миттєвої підтримки різняться за рядом параметрів, таких як швидкість відгуку, наявність інноваційних функцій, і можливостями інтеграції. На сьогоднішній день значення мають не лише базові функції, але й додаткові можливості, такі як персоналізація і автоматизація обслуговування, які покращують загальний досвід взаємодії з клієнтом [3].

І хоча деякі платформи пропонують розширені функції штучного інтелекту, вони можуть бути складними у налаштуванні та інтеграції, що створює бар'єри для впровадження, особливо

для малих та середніх підприємств. Ціновий фактор також відіграє важливу роль, особливо для стартапів та SMB, для яких доступність рішень є критичною.

Розглядаючи ці існуючі рішення, можна зробити висновок, що ключ до успіху сучасної системи миттєвої підтримки полягає у збалансуванні між потужними функціональними можливостями та простотою використання. Інвестиції у розвиток штучного інтелекту, машинного навчання та автоматизованих відповідей можуть значно покращити ефективність системи, але необхідно також звернути увагу на вартість інструментів та їхню доступність для різних сегментів ринку. У своєму розробленні слід враховувати ці аспекти, а також створити рішення, яке буде легко інтегрувати і масштабувати в залежності від потреб бізнесу.

Вимоги до системи на основі аналізу аналогів. Проведення детального аналізу існуючих систем миттєвої підтримки дозволяє визначити напрямки, на які слід зосередитися під час розробки нової системи. Основні аспекти, які необхідно врахувати, включають в себе інтеграційну сумісність, вартість впровадження, легкість використання, можливість масштабування та впровадження інноваційних технологій.

Інтеграційна сумісність – Система має легко інтегруватися з різними CRM-системами та іншими інструментами, які вже використовуються на підприємстві. Це дозволить забезпечити безперебійний потік інформації та уникнути дублювання даних.

Вартість впровадження – Важливо, щоб система була доступною для широкого спектру бізнесів, включаючи малі та середні підприємства. Передбачення гнучких тарифних планів та безкоштовного доступу до базових функцій може збільшити привабливість продукту на ринку.

Легкість використання – Інтуїтивно зрозумілий інтерфейс, що не вимагає складного навчання для операторів та кінцевих користувачів, є ключовим для швидкого впровадження та прийняття системи споживачами.

Можливість масштабування – Система повинна бути гнучкою, щоб витримувати збільшення обсягів запитів та користувачів без втрати продуктивності. Це дозволить бізнесу рости без необхідності частих оновлень системи.

Впровадження інновацій – Використання штучного інтелекту для автоматизації відповідей та аналітики даних дозволить не тільки знизити навантаження на операторів, але й забезпечити більш персоналізований підхід до кожного клієнта.

Безпека та конфіденційність – З огляду на зростаючу увагу до питань захисту персональних даних, система має включати надійні засоби захисту інформації та відповідати вимогам регуляторів [4].

На основі цих напрямків можна скласти технічне завдання для розробки системи, яка відповідатиме сучасним вимогам ринку та запитам користувачів. Розробка має бути спрямована на створення продукту, що пропонує зручне і ефективне вирішення проблем клієнтів, при цьому забезпечуючи високу якість обслуговування та унікальність користувацького досвіду.

Таким чином, стратегія розробки включатиме створення модульної платформи з можливістю легкої інтеграції у вже існуючі системи бізнесу, фокусування на оптимізації вартості для забезпечення доступності для всіх категорій бізнесів, а також інтеграції передових технологій для поліпшення якості обслуговування та персоналізації спілкування.

Метою роботи є розробка системи миттєвої підтримки користувачів з використанням сучасних веб-технологій, дослідження можливостей впровадження системи, її функціональних можливостей та переваг перед існуючими рішеннями.

Виклад основного матеріалу дослідження

Вибір технологій розробки. Вибір технологічного стеку для дослідження, що фокусується на розробці системи миттєвої підтримки, базується на сучасних практиках розробки програмного забезпечення [5]. Нижче наведений огляд основних технологій та бібліотек, що були обрані для реалізації проекту.

HTML: Мова програмування HTML використовується для структурування веб-сторінок, визначає, як буде виглядати контент на сторінці, такий як текст, зображення, відео тощо.

CSS: Стильова мова CSS застосовується для оформлення веб-сторінок, забезпечуючи їх привабливий та сучасний вигляд. Використання передпроцесорів, таких як Sass або LESS, дозволяє підвищити ефективність стилізації компонентів.

JavaScript: Як основна мова для розробки клієнтської частини, JavaScript використовується для створення інтерактивних елементів інтерфейсу та забезпечення асинхронної взаємодії з сервером за допомогою Ajax без необхідності повного перезавантаження веб-сторінки [6-8]. Динамічне змістове наповнення, анімації, валідація форм – усе це реалізовано на JavaScript.

PHP: Серверна мова програмування PHP задіяна для обробки запитів від користувачів, взаємодії з базою даних і виконання логіки бізнес-операцій. PHP було обрано через його широку підтримку хостинговими провайдерами та гнучкість у розробці веб-додатків [9, 10].

Hack: Мова програмування Hack використовується як модернізований діалект PHP, який надає додаткові можливості для типізації та обробки асинхронного коду, покращуючи продуктивність та безпеку застосунку.

Dockerfile: Конфігурація Docker дозволяє створювати ізольоване середовище для додатку, що спрощує розгортання та забезпечує однакові умови роботи додатку на різних машинах.

Бібліотека orhanerday/open-ai: Це клієнт API для OpenAI GPT-3, написаний на PHP, який інтегрується у систему, дозволяючи використовувати потужні алгоритми машинного навчання для автоматизації відповідей системи підтримки та обробки запитів користувачів на природній мові.

Завдяки цим технологіям та інструментам, розроблювана система отримує ряд переваг, які включають швидкість, безпеку, адаптивність і можливість інтеграції. Цей комплексний підхід забезпечує високу продуктивність та гнучке налаштування системи відповідно до змінних вимог користувачів та технологічних трендів.

Застосування бібліотеки orhanerday/open-ai в проєкті дозволяє використовувати переваги однієї з найбільш передових технологій в галузі штучного інтелекту, GPT-3 від OpenAI, що надає можливість обробляти природну мову на високому рівні. Використання цієї бібліотеки забезпечує розроблюваній системі здатність глибоко розуміти запити користувачів, пропонувати релевантні відповіді та вивчати взаємодії для покращення подальшого сервісу.

Використання Docker у проєкті гарантує, що програмне забезпечення може бути легко розгорнуте в будь-якому середовищі, що забезпечує консистентність між розробкою та виробництвом та відповідність сучасним практикам DevOps.

СКБД SQLite, хоча і зазвичай використовується в менших проєктах або для розробки прототипів, є вбудованою СКБД, яка не вимагає окремого сервера баз даних, що робить її популярною для додатків, що працюють в обмежених умовах, таких як мобільні додатки [11-13].

Також, важливо відзначити, що *безпека є основоположним елементом* у всіх аспектах розробки. Використання HTTPS, SSL-шифрування, сучасних алгоритмів аутентифікації та авторизації, а також систем захисту від SQL-ін'єкцій та XSS-атак, гарантує захист персональних даних користувачів та інтегритет системи [14, 15].

Разом, ці технологічні рішення складають надійний фундамент для розробки системи миттєвої підтримки, здатної задовольнити найсудовіші вимоги користувачів та бізнесу.

Проектування архітектури системи. Архітектура системи миттєвої підтримки втілює в собі гармонійну взаємодію між людиною та машиною, орієнтовану на забезпечення високоякісної та ефективної служби підтримки. Узявши за основу надану схему, можна побачити, як звичайний запит користувача перетворюється в діалог, у якому штучний інтелект та оператор працюють разом для досягнення кінцевої мети – задоволення потреб клієнта (рис. 1).

На першому етапі клієнт ініціює взаємодію з системою, висловлюючи своє запитання або заяву. Ця інформація збирається та структурується через інтерфейс користувача [16, 17], який розроблено таким чином, щоб бути інтуїтивно зрозумілим та простим у використанні. Потім,

дані надсилаються до сервера, де вони зберігаються у базі даних. Це дозволяє системі швидко отримувати доступ до історії користувача та до поточного контексту запиту.

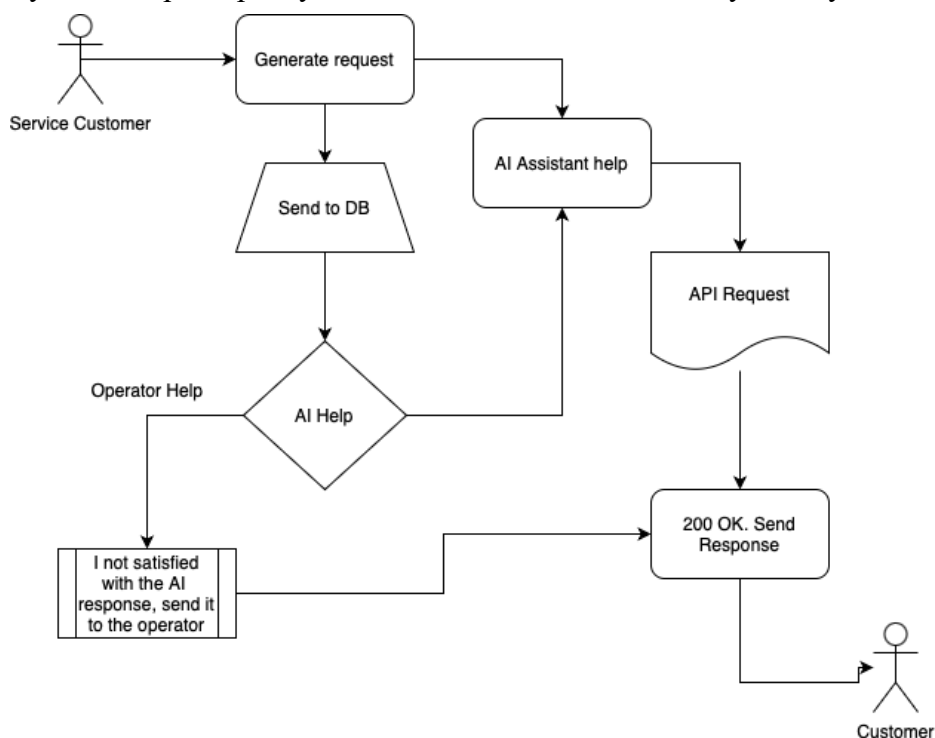


Рис. 1. Діаграма роботи застосунку

Наступний етап передбачає залучення модуля штучного інтелекту, який обробляє отримані дані та намагається надати найкраще можливе рішення. Штучний інтелект використовує попередньо накопичені дані, аналітику поведінки та алгоритми машинного навчання для того, щоб зрозуміти запит користувача і запропонувати відповіді.

У разі, якщо AI не знаходить відповіді, або вона не влаштовує користувача, система перенаправляє запит до живого оператора, який вже має всю історію діалогу та аналіз AI перед очима, що дає йому змогу більш ефективно вирішити проблему. У процесі обробки запиту може знадобитися інтеграція з внутрішніми та зовнішніми API для отримання додаткових даних або для виконання певних дій. Це вимагає створення безпечних та надійних каналів зв'язку між системою підтримки та іншими сервісами. Завершальний етап полягає в тому, що система надає відповідь користувачу. Це може бути просте повідомлення про успішне виконання запиту або, у випадку більш складних сценаріїв, детальна інструкція або рекомендація. За кожною відповіддю стоїть аналітика даних, що дозволяє системі не тільки відповісти на запит, а й збагатити внутрішню базу знань для покращення подальшого обслуговування.

Враховуючи необхідність гнучкості у взаємодії з клієнтом, система також забезпечує функціонал переходу від штучного інтелекту до живого оператора. Це означає, що в архітектурі системи передбачено не тільки алгоритми AI, а й засоби для управління сесіями в чаті, розподілу навантаження між операторами та ведення статистики ефективності роботи служби підтримки.

При проектуванні архітектури також слід звернути увагу на забезпечення високого рівня безпеки. Компонент безпеки включає шифрування даних, регулярне оновлення сертифікатів, захист від зовнішніх загроз і зловмисних атак, а також забезпечення відповідності вимогам законодавства щодо захисту даних.

Збалансоване поєднання цих компонентів і принципів в архітектурі створює міцну основу для системи миттєвої підтримки, яка здатна адаптуватися до різних сценаріїв використання та розширюватися разом зі зростанням потреб бізнесу.

Екранні форми системи. На рисунках 2-7 наведені екранні форми роботи системи:

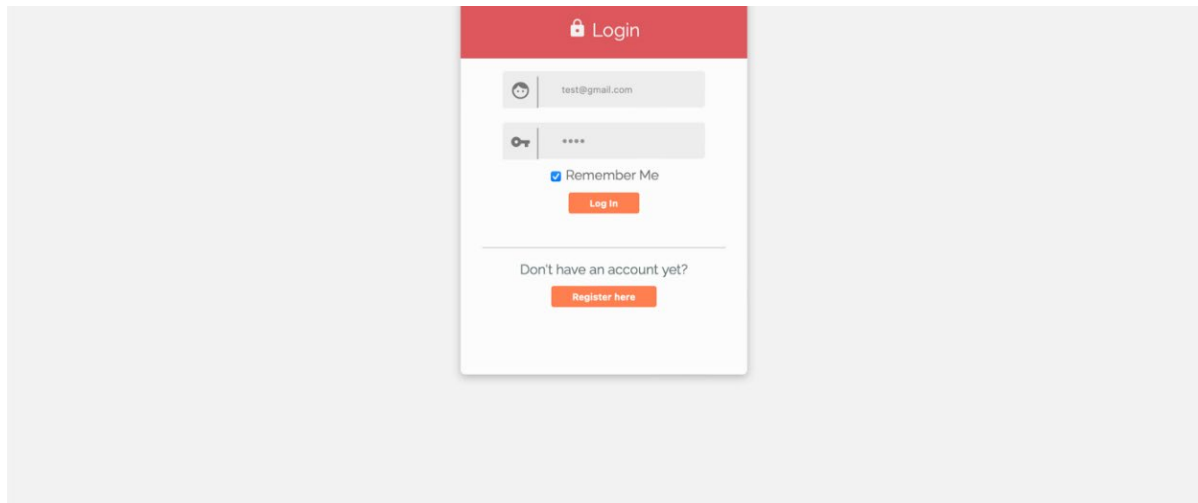


Рис. 2. Вікно ідентифікації

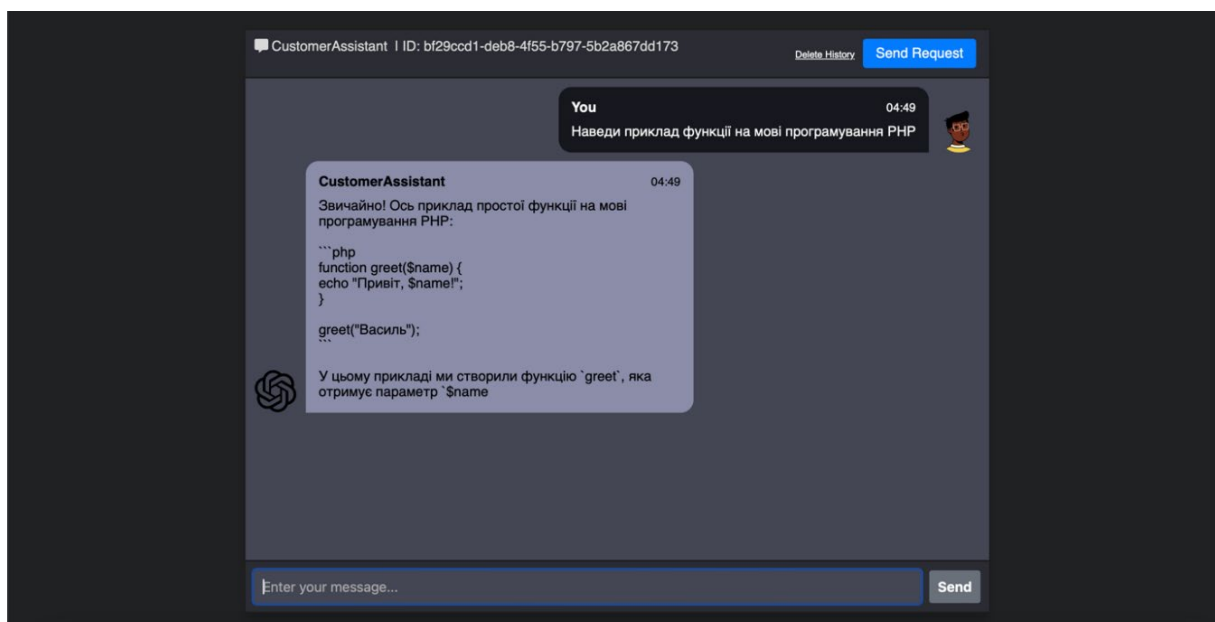


Рис. 3. Вікно із запитом та відповіддю

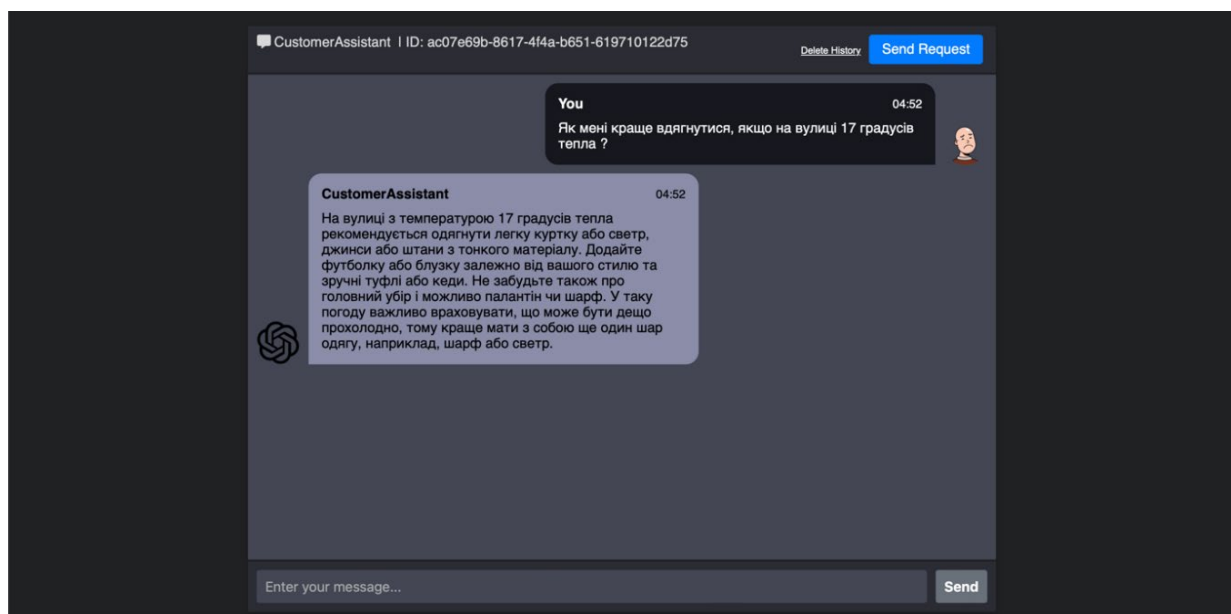


Рис. 4. Вікно із запитом та відповіддю

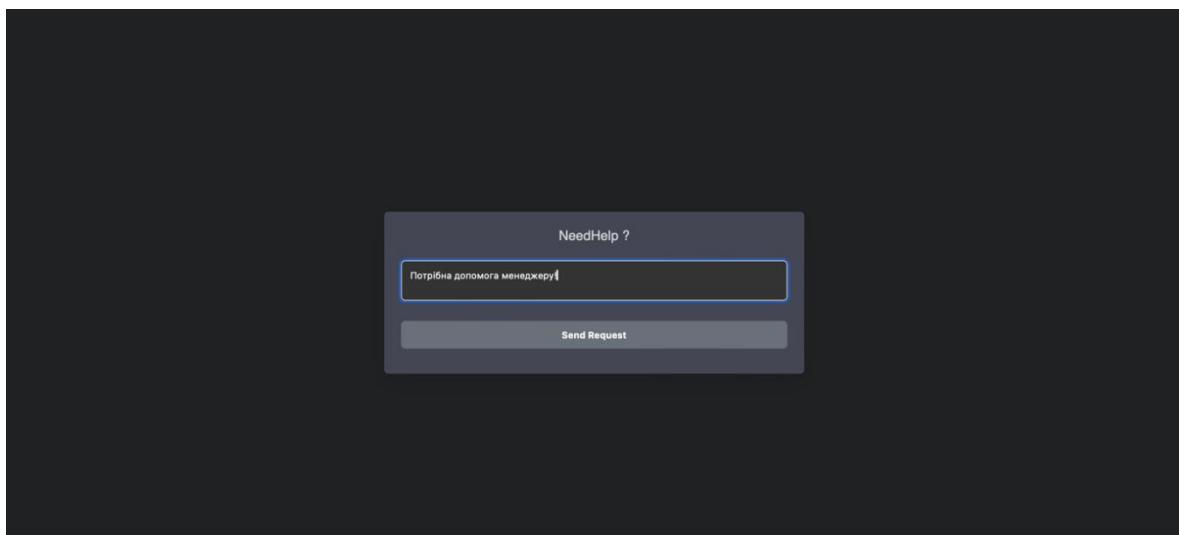


Рис. 5. Вікно підтримки

Support Requests

User ID	Message	Creation Date	Actions
1	23	2024-04-09 22:35:53	Reply
1	1231232	2024-04-09 22:39:20	Reply
1	Test message	2024-04-09 22:46:12	Reply
1	I want to speak with real person!	2024-04-09 22:51:44	Reply
1	Потрібна допомога менеджеру!	2024-04-10 01:53:36	Reply

Рис. 6. Вікно запитів користувача

id	name	email	password
1	Alex	test@gmail.com	\$2y\$10\$XK6QRucz...
2	test2	test2@gmail.com	\$2y\$10\$Esp5ZBRI...
3	TestSupport	support@website.c...	\$2y\$10\$6z8m65Ye...

Рис. 7. Підсумкове вікно запитів

Висновки

В статті було проведено огляд і аналіз існуючих систем миттєвої підтримки, аналіз потреб та вимог користувачів щодо миттєвої підтримки. В результаті аналізу були визначені ключові функції та можливості, які мали б бути доступні в системі.

Спроектowana та розроблена система миттєвої підтримки, яка базується на мові програмування PHP та використанні технології Ajax для забезпечення швидкості та інтерактивності. Використання цих технологій дозволило забезпечити високу швидкість обробки запитів та надання відповідей без потреби у перезавантаженні сторінки, що суттєво підвищує зручність користування системою.

Особливу увагу було приділено процесам оптимізації та тестування системи, завдяки чому вдалося значно підвищити її продуктивність. Покращення ключових показників ефективності, таких як час відгуку, пропускна спроможність та використання ресурсів сервера, підтвердили високу ефективність реалізованих оптимізаційних заходів.

Розроблена система миттєвої підтримки має великий потенціал для використання у різних галузях, де необхідно забезпечити швидке та якісне обслуговування користувачів.

Можливості подальшого розвитку та удосконалення системи, зокрема через інтеграцію з розширеними алгоритмами штучного інтелекту та розробку адаптивного інтерфейсу, відкривають нові перспективи для покращення якості обслуговування та задоволення потреб користувачів.

Результати даної роботи будуть корисні для компаній, які надають онлайн-сервіси та продукти, оскільки ефективна система миттєвої підтримки дозволить підвищити задоволеність користувачів та зменшити час вирішення їхніх проблем.

Список використаної літератури:

1. Stuttard D., Pinto M. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. NY: Wiley, 2020. 912 с.
2. Ballad T., Confer W. Securing PHP Web Applications. MA: Syngress, 2008. 304 с.
3. Snyder C., Southwell M., Owad T. PHP Security. CA: O'ReillyMedia, 2005. 428 с.
4. Barnett R.C. Preventing Web Attacks with Apache. NJ: Pearson Education, 2006. 448 с.
5. Schlossnagle G. Advanced PHP Programming. NJ: Pearson Education, 2004. 224 с.
6. Bergmann S., Pribsch S. Foundations of PHP for Web Developers. NY: Apress, 2015. 372 с.
7. Zandstra M. PHP Objects, Patterns, and Practice. NY: Apress, 2013. 536 с.
8. Nixon R. Learning PHP, MySQL & JavaScript. CA: O'Reilly Media, 2018. 832 с.
9. McDonald M., McGovern J., et al. Web Security for Developers: Real Threats, Practical Defense. CA: O'Reilly Media, 2022. 542 с.
10. Wenz C. PHP Phrasebook. IN: Addison-Wesley, 2005. 480 с.
11. Nixon R. Modern PHP: New Features and Good Practices. CA: O'ReillyMedia, 2015. 268 с.
12. Welling L., Thomson L. PHP and MySQL Web Development. CA: Addison-Wesley, 2017. 672 с.
13. Duckett J. PHP & MySQL: Novice to Ninja. UK: SitePoint, 2017. 690 с.
14. Shiflett C. Essential PHP Security. CA: O'Reilly Media, 2005. 100 с.
15. Sklar D., Trachtenberg A. PHP Cookbook. CA: O'Reilly Media, 2014. 820 с.
16. Bergmann S., Pribsch S. PHPUnit Pocket Guide. CA: O'ReillyMedia, 2005. 120 с.
17. Ullman L. PHP and MySQL for Dynamic Web Sites. CA: PeachpitPress, 2017. 696 с.

Автори статті

Шикун Олена – доктор фізико-математичних наук, професор, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0000-0002-7385-2816

Іщераков Сергій – кандидат технічних наук, доцент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0007-5961-8218

Савончук Денис – студент, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0009-0935-4888

Антонов Віталій – аспірант, Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0009-0003-6664-5282

Authors of the article

Shykula Olena – Doctor of Sciences (physics and mathematics), Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0000-0002-7385-2816

Ishcheryakov Sergii – Candidate of Sciences (technical), Associate Professor, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0007-5961-8218

Savonchuk Denys – student, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0009-0935-4888

Antonov Vitaly – postgraduate, State University of Information and Communication Technologies, Kyiv, Ukraine.

ORCID: 0009-0003-6664-5282