Семенов А.Р., магістрант
Донбаська державна машинобудівна академія

ЩОДО ОРГАНІЗАЦІЙНИХ ШЛЯХІВ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ РОБОТИ IT-КОМАНДИ

Ефективна комунікація у сучасному і динамічному робочому середовищі є важливою для результативної співпраці та згуртованості мережевих команд. Оскільки дистанційна робота та гнучкий графік з численними фактичними накладеннями графіків та розкладу стають все більш поширеними реаліями, то компанії дедалі більше покладаються на гнучкі хмарні цифрові рішення, які здатні ефективно підтримувати належний рівень ділової комунікації між дистанційно працюючими співробітниками, де б вони фізично не знаходилися. Мережеві платформи, веб-додатки та застосунки для здійснення відеодзвінків та відеоконференцій є одним із інструментів, що надають можливість для робочого спілкування в режимі реального часу, а також подолання додаткового соціально-психологічного навантаження, спричиненого фізичною відстанню, як між окремими працівниками, так і між цілими мережевими колективами, віддалено працюючими в сучасній IT сфері.

Підвищення продуктивності діяльності міжнародної IT команди передбачає не лише регулярну аудіо-візуальну-комунікацію в рамках проведення численних корпоративних відео-конференцій (із можливим застосуванням сучасних методів штучного інтелекту (ШІ) для подальшої автоматичної генерації субтитрів до записаних стрімів), але і одночасну роботу над поточною версією програмного коду (приміром в рамках спільного використання таких відкритих репозиторіїв як GitHub [1], Stack Overflow [2] та ін.).

З міркувань тайм менеджменту працівникам рекомендується знаходити додатковий час на регулярне коротке планування очікуваної робочої активності шляхом додавання поточних планів, які в подальшому зберігаються в записах загальної бази даних SQL-типу.

З точки зору автора дослідження, задля підвищення продуктивності роботи команди та забезпечення належного рівня комунікаційної конфіденційності розробника, було б цікаво та корисно запропонувати інтегровану веб платформу, яка б одночасно забезпечувала як інформаційно-комунікативні можливості для здійснення відеодзвінків, так і ефективну реалізацію функціоналу мережевого репозиторію разом із планувальником поставлених задач [3].

Запропонована інтегрована платформа для спільної роботи девелоперів повинна забезпечити досягнення наступних переваг:

– Спрощене спілкування та ефективна співпраця розробників;
– Заощадження робочого часу шляхом зменшення числа переключень між різними вікнами активних застосунків;
– Більшу централізацію та доступність технічної документації.

Для розробки Frontend-частини запропонованої платформи були використані Веб-технології Vue.js, Node.js, Socket.IO, PeerJS. Backend-частину було написано на Python, Django та SQLite.

Реєстрація нового користувача додатку дозволяє створення та подальшу авторизацію у особистому профілі на платформі.

Перший компонент дизайну запропонованого додатку відповідає за здійснення відеозв’язку. Користувач ініціює відеодзвінок через HTTP-запит до Vue.js. Далі, за допомогою Node.js і Socket.IO встановлюється WebSocket-з'єднання для обміну сигналізацією та створення P2P-з'єднання через Peer.js. Спеціальний вузол обробляє відеопотік та відправляє його назад до браузера користувача як HTML-сторінку.

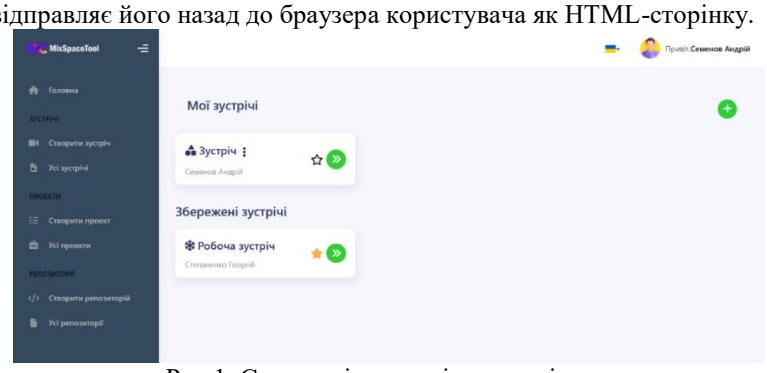

Рис.1. Список кімнат з відеозустрічами

Користувач застосунку має змогу щоразу створювати запрошення багатьох учасників до однієї поточно-активної власної кімнати для проведення відео-конференції, причому кожна з можливих конференційних кімнат не лише відповідає певному унікальному покликанню, але є єдино-активною конференцією в даний момент часу, оскільки реалізація паралельних конференційних кімнат не планувалася. Користувач застосунку також може приєднуватися до кімнат інших учасників, попередньо зареєстрованих на запропонованій платформі. Підключення до кімнати відео-конференції є можливим після одержання запрошення зі сторони ініціатора зустрічі, причому одержане запрошення відображається в групі записів “Збережені зустрічі”. Для підвищення рівня безпеки, після завершення важливої конфіденційної розмови, один з користувачів платформи може видалити і запис про зустріч і лінк зустрічі з платформи (Рис.1).

Другий компонент відповідає за репозиторій програмного забезпечення. Для роботи сховища використовуємо стек технологій Vue.js, Node.js, Django і SQLite для забезпечення взаємодії користувача з мережевим репозиторієм. Vue.js слугує як односторінковий додаток, Node.js забезпечує серверну частину для

Vue.js, Django відповідає за обробку інформації та взаємодію з базою даних SQLite. Дані передаються між компонентами системи у форматі JSON. Репозиторій зберігається у базі даних, а його файли і теки окремо на сервері.

Інтегрована опція контролю версій, у разі виникнення проблем із поточною версією, дозволяє повернутися до старої. Такий процес у додатку реалізовано за допомогою комітів, які роблять своєрідний знімок поточного стану проекту, включаючи зміни в коді, конфігураційних файлах та документації. Присутня можливість додавання до свого репозиторію інших користувачів, для кооперативної розробки програмного забезпечення. Існує здатність виконувати повний цикл роботи з файлами: від створення нових тек і файлів до їх перейменування та видалення. Також передбачена можливість завантажувати необхідні файли з зовнішніх джерел або вивантажувати їх для подальшого використання (Рис.2).

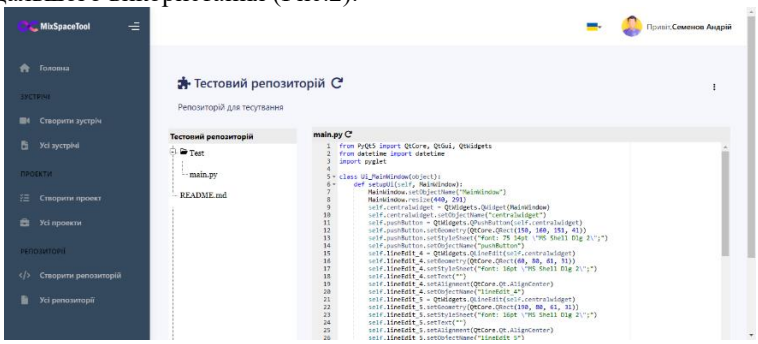

Рис.2. Репозиторій з програмним кодом

Третій компонент представляє собою планувальник завдань для працівників. Взаємодія з планувальником завдань здійснюється через веб-інтерфейс: HTTP-запити передаються до клієнтської частини, розробленої на Vue.js та розміщеної на сервері Node.js до Django, який формує SQL-запити до SQLite. Результати в форматі JSON повертаються через Django на клієнт, де генерується HTML-сторінка для браузерного відображення.

Є можливість самостійно створювати доручення і залучати туди інших працівників. Користувач має змогу додати опис завдань для більш докладного роз'яснення, також можливо вказати пріоритет і вибрати колір, що буде сигналізувати про їх важливість. Якщо поставлена задача була виконана, користувач може зазначити, що завдання завершене. Додатково вбудовано таймер, що дозволяє відстежувати свій прогрес роботи над проектом (Рис.3).

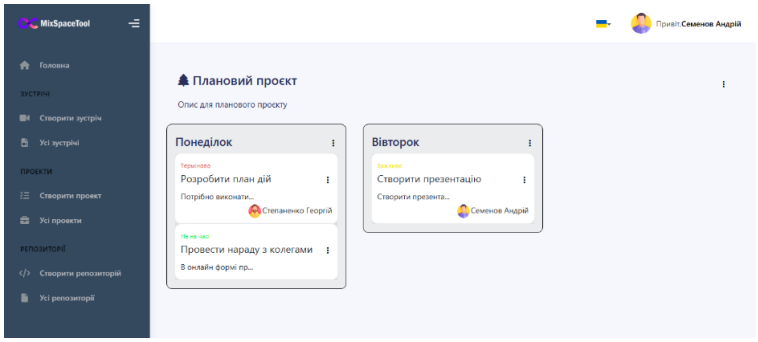

Рис.3. Список запланованих завдань у менеджері по задачах

Автор вважає, що запропонований проект інтегрованої платформи буде зручним засобом для командної роботи розробників. Використання запропонованого рішення сприятиме співпраці учасників проекту. Інтеграція необхідного інструментарію сприятиме можливості оптимізації співпраці між розробниками, що прискорить процеси командного створення нового програмного забезпечення.

На думку автора, запропоноване рішення також може мати педагогічні застосування в рамках розширення спектру індивідуальних студентських завдань та удосконалення лабораторно-практичного компоненту в межах покращення методики викладання таких прикладних дисциплін, як веб-програмування та веб-технології.

Список використаних джерел

1. GitHub Build and ship software on a single, collaborative platform. URL: https://github.com (дата звернення: 23.11.2024).
2. Stack Overflow – Where Developers Learn, Share, & Build Careers. URL: https://stackoverflow.com/ (дата звернення: 23.11.2024).
3. GitHub – Semenov-Andrii/MixSpaceTool. URL: https://github.com/Semenov-Andrii/MixSpaceTool (дата звернення: 23.11.2024).