

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему:

«Розробка Front End частини платформи для конференцій на Angular
фреймворку»

Виконала: студентка 4 курсу, групи ПД-44
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вероївська А.О.

(прізвище та ініціали)

Керівник Корецька В.О.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти – «Бакалавр»

Напрямок підготовки – 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В

“ _____ ” _____ 2022 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

_____ Вероївській Анастасії Олексіївні _____

(прізвище, ім'я, по батькові)

Тема роботи: Розробка Front End частини платформи для конференцій на Angular фреймворку

Керівник роботи Корецька Вікторія Олександрівна, кандидат педагогічних наук, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “18” лютого 2022 року №22.

1. Строк подання студентом роботи 03.06.2022

2. Вхідні дані до роботи:

- 2.1 Офіційна документація Angular
- 2.2 Figma
- 2.3 Web Storm
- 2.4 Scrum_

3. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- 3.1 Аналіз актуальності та проблематики розроблюваного додатку.
- 3.2 Аналіз та вибір інструментів для реалізації веб додатку.
- 3.3 Проектування веб додатку.

3.4

3.5 Висновки

4. Перелік графічного матеріалу

4.1 Титульний слайд

4.2 Мета, об'єкт, предмет та наукова новизна дослідження

4.3 Актуальність роботи

4.4 Аналоги

4.5 Технічні завдання

4.6 Програмні засоби реалізації

4.7 Зв'язок елементів додатку

4.8 UML діаграм архітектури веб додатку

4.9 Апробація результатів дослідження

4.10 Висновки

4.11 Кінцевий слайд_____

5. Дата видачі завдання 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми бакалаврської роботи	11.04 - 14.04	Виконано
2	Підбір науково-технічної літератури	15.04 - 17.04	Виконано
3	Дослідження аналогів та актуальності додатку	18.04 - 21.04	Виконано
4	Аналіз та вибір інструментів для розробки додатку	22.04 - 25.04	Виконано
5	Проектування та реалізація	26.04 – 05.05	Виконано
6	Висновки, оформлення роботи	05.05 – 07.05	Виконано
7	Розробка демонстраційних матеріалів	07.05 – 10.05	Виконано
8	Розробка обов'язкових демонстраційних матеріалів	11.05 – 15.05	Виконано
9	Попередній захист роботи	16.05 – 01.06	Виконано
10	Здача роботи	03.06	

Студент

(підпис)

Вероївська А.О.

(прізвище та ініціали)

Керівник роботи

(підпис)

Корецька В.О.

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 60 с., 1 табл., 36 рис., 9 джерел

Ключові слова: JavaScript, Angular, RxJs, Figma, GIT, GitFlow, Karma.

Об'єкт дослідження – процес проведення конференцій.

Предмет дослідження – сучасні технології в процесі проведення конференцій.

Мета дослідження – покращення процесу проведення веб конференцій за допомогою розробленого веб додатку для конференцій на Angular фреймворку.

Методи дослідження – методи теорії інформації, методи оптимізації, методи оптимального управління, методи порівняння, аналіз отриманих результатів.

У дипломній роботі був проведений аналіз аналогів що вже існують, переваг та недоліків програмних інструментів для веб додатка.

Виконано опис програм, які були використані у ході роботи, та середі розробки.

На основі результатів виконаних досліджень розроблено веб додаток. Додаток реалізований за допомогою мови програмування JavaScript, з використанням фреймворку Angular, для кросплатформної розробки веб додатку.

ЗМІСТ

РЕФЕРАТ	6
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	10
ВСТУП.....	11
РОЗДІЛ 1	12
ТЕОРЕТИЧНА ЧАСТИНА.....	12
1.1 Загальні уявлення про веб-конференції	12
1.2 Види конференцій	13
1.2.1 Веб-трансляція	13
1.2.2 Вебінар.....	14
1.2.3 Веб-зустріч	15
1.2.4 Відеоконференція	16
1.2.5 Гібридні моделі.....	16
1.3 Актуальність проведення веб-конференцій	17
1.3.1 Для чого використовувати веб-конференції.....	17
1.3.2 Гостьовий доповідач	18
1.3.3 Робочий час або віртуальні сесії огляду	18
1.3.4 Простір групової співпраці.....	19
1.3.5 Онлайн клас	19
1.4 Аналіз наявних додатків	19
1.5 Складання технічного завдання	21
1.5.1 Структура сайту.....	21
1.5.2 Прототип	23

РОЗДІЛ 2	24
ОПИС ПРОГРАМНИХ ЗАСОБІВ	24
2.1 Мова програмування JavaScript	24
2.1.1 Як JavaScript робить речі динамічними	25
2.2 Мова програмування TypeScript	25
2.2.1 TypeScript досягає своїх цілей трьома способами:	26
2.3 Angular	27
2.3.1 Основні відмінності між JavaScript і Angular	28
2.4 RxJs	30
2.5 Figma	32
2.6 Протоколи взаємодії HTTP\HTTPS	34
2.7 Rest API	36
2.8. Переваги засобів автоматизації тестування над ручним	38
2.8.1 Що таке автоматичне тестування?	38
2.8.2 Переваги автоматизованого тестування	38
2.8.3 Види засобів автоматизації тестування	40
2.8.4 Життєвий цикл автоматизації тестування	40
2.9 Karma	41
РОЗДІЛ 3	44
ЗАГАЛЬНІ ВІДОМОСТІ ДОДАТКУ	44
3.1 Розробка концепції	44
3.2 Проектування структури та створення додатку	45
ВИСНОВКИ	60

ПЕРЕЛІК ПОСИЛАНЬ	62
ДОДАТОК А	63

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

JS – мова програмування JavaScript

BBC – British Broadcasting Corporation

CNN – Cable News Network

VoIP – Voice over Internet Protocol

URL – Uniform Resource Locator

CSS – Cascading Style Sheets

HTML – Hyper Text Markup Language

DOM – Document Object Model

Rx – Reactive Programming

HTTP – Hypertext Transfer Protocol

ВСТУП

На сьогоднішній день розвитку інформаційних технологій, все більше людей користуються конференц-зв'язками для спілкування одночасно з багатьма людьми.

Об'єкт дослідження – процес проведення конференцій.

Предмет дослідження – сучасні технології в процесі проведення конференцій.

Мета дослідження – покращення процесу проведення веб конференцій за допомогою розробленого веб додатку для конференцій на Angular фреймворку.

Наукова новизна проекту – створення унікального веб додатку для контролю проведення конференцій з використанням системи односторінкового веб додатку та можливістю відстеження нових сесій.

У дипломному проекті був проведений аналіз додатків-аналогів, та виявлені переваги та недоліки.

Додаток було розроблено на мові програмування JavaScript з використанням Angular Framework у середовищі розробки Web Storm для платформи Web. Інтерфейс проектувала у сервісі Figma. Для тестування використовувала - Karma. Додаток може використовуватись для самоконтролю.

Сучасний розвиток інформаційних технологій дає можливість розширення спілкування та навчання в режимі онлайн. Все більше людей користуються конференц-зв'язками для спілкування одночасно з декількома людьми. Проведення веб-конференцій можуть сприяти в онлайнному, змішаному або традиційному навчанні.

Веб-конференції – це простий спосіб раціонального використання робочого часу. Тому надзвичайно актуальним є розробка веб додатку для проведення конференцій що сприятиме самоконтролю та мотивації користувача в навчальному процесі, покращить навички комунікації у колективі.

РОЗДІЛ 1

ТЕОРЕТИЧНА ЧАСТИНА

1.1 Загальні уявлення про веб-конференції

Веб-конференція — це будь-який тип зустрічі, в якій беруть участь два або більше учасників у різних місцях.

Ці учасники можуть перебувати у будь-якій точці світу. Завдяки надійному інтернет-з'єднанню та програмному забезпеченню для проведення конференцій вони можуть бачити, говорити та чути один одного в режимі реального часу.

Це програмне забезпечення для відеоконференцій у режимі реального часу, яке дозволяє вам бачити, чути та віртуально взаємодіяти з людьми за допомогою комп'ютера, планшета або мобільного пристрою.

У той час як веб-конференції існують уже кілька років — «скайпінг» зараз є у багатьох словниках — їхнє освітнє використання та вплив лише починають повністю вивчатися. Тепер це можна вважати технологією навчання, яку можна використовувати у будь-якому середовищі класу, що полегшує викладання та навчання за рахунок кращої взаємодії викладачів та студентів.

Розвиток технології веб-конференцій дає людям можливість проводити зустрічі в режимі реального часу, незважаючи на національні кордони та географічні відстані. Це, по суті, допомагає зблизити людей всього світу, дозволяючи їм вирішувати питання, що становлять спільний інтерес. Крім того, у разі стихійного лиха, порушень прав людини, екологічних порушень та інших нагальних проблем ця технологія дозволяє людям у різних частинах світу швидше знаходити спільні засоби для вирішення таких проблем.

Веб-конференції варіюються від простого спілкування двох людей зі своїх домашніх офісів до зустрічей членів правління та/або співробітників багатомільйонних корпорацій та великих асоціативних конференцій із тисячами учасників. Серед функцій, які зазвичай використовуються на зборах, - презентації

у вигляді слайд-шоу, живе або потокове відео, аудіозв'язок у реальному часі, загальнодоступний або приватний текст серед учасників, підключених до зборів, торгові виставки, мережеві заходи, опитування та огляди, а також постерні сесії. Спільне використання екрана може дозволити учасникам бачити все, що доповідач показує на своєму екрані.

Об'єднуючи людей у мережі, веб-конференції дозволяють економічно проводити зустрічі між людьми, які перебувають у різних місцях, без витрат часу та коштів на фізичні зустрічі, коли всі збираються в одному місці. Спільне використання файлів, спільний перегляд веб-сторінок та спільне використання програм також легко підтримуються через Інтернет. Внаслідок ефективності цього методу все більше покладаються на веб-конференції замість традиційних особистих зустрічей.

1.2 Види конференцій

Існує три різні типи веб-конференцій:

- веб-трансляція;
- вебінар;
- веб-нарада.

Однак між цими термінами є значне збігання.

1.2.1 Веб-трансляція

Веб-трансляція — це онлайн-трансляція, за допомогою якої аудіо- або відеовміст з одного джерела передається через Інтернет багатьом слухачам/глядачам за допомогою технології потокового медіа. Медіа-файл може розповсюджуватися наживо або на вимогу. Термін веб-трансляція зазвичай зарезервований для позначення неінтерактивних лінійних потоків або подій. Сотні

слухачів/глядачів можуть переглядати веб-трансляцію одночасно, але взаємодія між цією аудиторією обмежена або відсутня.

Веб-трансляція — це «інтернет-аналог традиційного радіо- і телемовлення», а найбільші «веб-трансляції» включають існуючі радіо- та телевізійні станції, які «одночасно передають» свій вихід, а також безліч «станцій» лише в Інтернеті. Практично всі основні телерадіокомпанії мають веб-трансляцію своєї продукції, від BBC до CNN, до Аль-Джазіри, до UNTV на телебаченні, до Радіо Китаю, Радіо Ватикану, Радіо Організації Об'єднаних Націй і Всесвітня служба на радіо.

Сьогодні веб-трансляціями користуються все частіше і початківці користувачі. Прямі веб-трансляції дають змогу переглядати презентації, ділові зустрічі, семінари тощо для тих, хто працює віддалено, а не відвідує. Такі сайти пропонують пряму трансляцію як доступне рішення для публічних виступів, яке розширює аудиторію для всіх, хто має підключення до Інтернету.

Спортивні події в прямому ефірі, як місцеві, так і національні, також швидко стали частими темами веб-трансляцій. Що стосується менших подій, таких як Мала ліга, аматорський спорт, спорт в невеликих коледжах та спорт середніх шкіл, вебтрансляція дозволяє таким подіям мати повне аудіо- або відеовисвітлення онлайн, коли вони не можуть забронювати стандартний час для радіо або телебачення. Такі веб-сайти, як Meridix Webcast Network, Texas Sports Radio Network та інші, дозволяють місцевим школам, командам і мовникам створювати власні вебтрансляції, які також мають перевагу в тому, що вони доступні будь-кому, хто має підключення до Інтернету (тобто родичам у кількох штатах), на відміну від діапазон і ринкові обмеження наземного радіо і телебачення.

1.2.2 Вебінар

Вебінар – це особливий тип веб-конференції, що включає лекцію або семінар, представлений через Інтернет. Вебінари, як правило, є односторонніми подіями від доповідача до аудиторії з обмеженою взаємодією з аудиторією (наприклад, у вебтрансляції). Однак цей термін також включає ті семінарські заходи, які мають

часткову або повну взаємодію між аудиторією та доповідачами, включаючи сесії запитань і відповідей.

У деяких випадках ведучий може говорити за стандартною телефонною лінією, вказуючи на інформацію, що відображається на екрані, а аудиторія може відповідати по своїх власних телефонах, які забезпечують найбільший комфорт і зручність. На ринку існують технології веб-конференцій, які включають використання аудіотехнології VoIP, щоб забезпечити повне спілкування через

Інтернет. Залежно від постачальника, вебінари можуть забезпечувати приховані або анонімні функції учасників, через що учасники не знають про інших учасників тієї ж зустрічі.

Термін вебінар, що поєднує веб і семінар, є неологізмом — нещодавно придуманим словом, яке, можливо, входить у загальне вживання, але ще не прийнято в основну мову. Термін «вебінар» був зареєстрований у 1998 році Еріком Р. Корбом (серійний номер 75478683, USPTO), але його було важко захистити; наразі він призначений для InterCall.

1.2.3 Веб-зустріч

Веб-зустріч – це особливий тип веб-конференції, у якій є високі можливості взаємодії між учасниками. На зустрічі двоє або більше людей збираються разом з метою обговорення (зазвичай заздалегідь визначеної) теми, часто в офіційній обстановці. У веб-конференції такі збори проводяться, коли кожен учасник сидить перед власним комп'ютером і підключається до інших учасників через Інтернет. У цю категорію входять засідання ради та співробітників, що проводяться через Інтернет, а також великі конференції з учасниками та доповідачами. Компанії з віддаленими філіями чи домашніми офісами вважають цей метод зручним для обговорення без поїздки в спільне місце, з учасниками, які можуть спілкуватися та обмінюватися документами в режимі реального часу.

1.2.4 Відеоконференція

Веб-конференція, яка використовує відео, може вважатися особливим типом відеоконференції, у якій вміст передається через Інтернет. Відеоконференції – це термін, який відноситься до відеосесії в режимі реального часу між двома або більше користувачами в двох або більше місцях. Ранні версії, до популярності Інтернету, використовували супутниковий зв'язок і традиційні аналогові телевізори.

Системи для будинків стали популярними на початку 1980-х років. Відеоконференції дозволяють спілкуватися між кількома кінцевими точками, тоді як терміни «відеочат» і «відеофон» зазвичай відносяться до комунікації один на один (PCMag 2010b).

1.2.5 Гібридні моделі

Гібридні моделі – це події, які проводяться віч-на-віч у певному місці, але частина або весь вміст також передається за допомогою веб-конференцій.

Інші типові особливості веб-конференції включають:

1. Презентації слайд-шоу – де слайди PowerPoint або Keynote представлені аудиторії, а інструменти розмітки та віддалений покажчик миші використовуються для залучення аудиторії, поки доповідач обговорює вміст слайду.
2. Відео в прямому ефірі або потокове відео – де веб-камера, цифрова відеокамера або мультимедійні файли передаються аудиторії.
3. VoIP (звуковий зв'язок у реальному часі через комп'ютер за допомогою навушників і динаміків)
4. Веб-тури – де URL-адреси, дані з форм, файли cookie, сценарії та дані сеансу можуть передаватися іншим учасникам, що дозволяє їм передавати їх за допомогою веб-входу, кліків тощо. Цей тип функції добре працює під час демонстрації веб-сайтів, на яких самі користувачі також можуть брати участь.

5. Запис наради – де активність презентації записується на стороні клієнта або сервера для подальшого перегляду та/або поширення.
6. Дошка з анотацією (що дозволяє доповідачеві та/або учасникам виділяти або позначати елементи на слайд-презентації. Або просто робити нотатки на чистій дошці).
7. Текстовий чат – для сеансів запитань і відповідей у прямому ефірі, обмежено для людей, підключених до зустрічі. Текстовий чат може бути відкритим (відповідається всім учасникам) або приватним (між двома учасниками).
8. Опитування та опитування (дозволяє ведучому задавати запитання з кількома варіантами відповідей, спрямованими на аудиторію)
9. Спільний доступ до екрана/спільний доступ до робочого столу/додатки (де учасники можуть переглядати все, що доповідач зараз показує на своєму екрані. Деякі програми для спільного використання екрана дозволяють керувати віддаленим робочим столом, дозволяючи учасникам керувати екраном доповідача, хоча це не використовується широко.).
10. Веб-конференції часто продаються як послуга, розміщена на вебсервері, що контролюється постачальником. Пропозиції різняться в залежності від постачальника, але більшість розміщених послуг передбачає модель вартості за користувача за хвилину, модель щомісячної фіксованої плати та модель місця. Деякі постачальники також пропонують рішення на стороні сервера, яке дозволяє клієнту розміщувати власну службу веб-конференцій на власних серверах.

1.3 Актуальність проведення веб-конференцій

1.3.1 Для чого використовувати веб-конференції

Існують різні способи, якими веб-конференції можуть допомогти в онлайн-овому, змішаному або традиційному очному курсі. Наприклад,

вебконференції дають простий спосіб утримувати робочий час, не перебуваючи в тому самому місці. Це стосується вашого курсу, незалежно від того, є він онлайн чи ні. Нижче наведено кілька випадків використання, які можуть стати в нагоді для з'ясування того, як веб-конференції можуть допомогти вам досягти ваших цілей викладання та навчання.

1.3.2 Гостьовий доповідач

Веб-конференції дозволяють легко залучити запрошеного доповідача на ваш курс, не турбуючись про матеріально-технічне забезпечення доставлення її до кампусу. Веб-конференції не тільки дозволяють студентам почути запрошеного доповідача, але й взаємодіяти та задавати запитання. Крім того, більшість програмного забезпечення для веб-конференцій дозволяє легко записувати сеанс і зробити його доступним для перегляду або для студентів, які не змогли встигнути в початковий час.

1.3.3 Робочий час або віртуальні сесії огляду

Ви можете легко використовувати веб-конференції для проведення особистих

або групових зустрічей зі своїми студентами. Більшість опцій дозволяють ділитися файлами або навіть екраном, щоб ви могли обговорювати матеріали курсу, оцінки та завдання. Веб-конференції також полегшують вашим студентам задавати вам запитання, показуючи вам конкретний вміст, щодо якого вони потребують роз'яснення. Виходячи з робочих годин один на один, ви також можете провести віртуальну сесію огляду з кількома студентами та записати її, щоб ті, хто не може бути присутнім, все одно могли отримати до неї доступ. Усе це дозволяє висвітлювати запитання студентів під час підготовки до завдання чи іспиту, не витрачаючи дорогоцінний час на урок.

1.3.4 Простір групової співпраці

Потрібен простий спосіб для учнів працювати над груповим завданням? Ви можете створювати сеанси веб-конференцій або кімнати, де студенти можуть бути модераторами. Це дозволяє студентам легко використовувати простір для роботи над груповим проектом або завданням або швидко зустрічатися з будь-яким, що стосується курсу. Студентам більше не потрібно шукати фізичну кімнату чи місце для всіх зустрічей. Це також чудовий спосіб об'єднати студентів у онлайн-курсі або об'єднати студентів різних класів — або навіть університетів — разом.

1.3.5 Онлайн клас

Ви можете провести лекцію або інтерактивне заняття повністю онлайн за допомогою веб-конференцій. Деяке програмне забезпечення навіть дозволяє створювати сеанси, щоб студенти могли працювати в менших групах, щоб відповісти на запитання або працювати над тематичним дослідженням. Потім ви можете повернути всіх до головної кімнати, щоб обговорити, над чим вони працювали. Слайдами для дошки можна обмінюватися між головною кімнатою та кімнатами для проведення зустрічей.

1.4 Аналіз наявних додатків

Zoom інтегровано з Canvas, і UW-Madison має корпоративний контракт з інструментом, що робить його рекомендованим варіантом для викладання та навчання.

Коли ви намагаєтеся вирішити, який із них використовувати, необхідно враховувати ряд факторів, таких як вартість, доступність, простота використання, юридична відповідальність та функціональність. Ми зібрали порівняльну таблицю доступних інструментів для веб-конференцій, яка включає Zoom, а також Microsoft

Teams, WebEx і Google Meet.

Таблиця 1 – Аналіз конкурентів

Особливості	Google Meet	Microsoft Teams	Zoom
Максимум учасників	100	10,000	1,000
Функція чату	Так	Так	Так
Можливості обміну файлами	Ні	Так	Так
Можливості спільного використання екрана	Так	Так	Так
Кімнати очікування	Ні	Так(50)	Так(50)
Веб версія	Так	Так	Так
Інтеграція живих/закритих субтитрів	Ні	Ні	Так
Студенти можуть створити сесію	Так	Так	Так
Пряма трансляція	Ні	Так	Так
Запис зустрічі	Ні	Так	Так
Зберігання запису в хмарі	Ні	Ні	Так

1.5 Складання технічного завдання

Технічне завдання – це документ із вимогами до сайту. Якщо ТЗ складено чітко та докладно, тоді будуть зрозумілі поставлені завдання.

1.5.1 Структура сайту

Структура – фундамент сайту. Її створення – найважливіший етап роботи.

Якщо вона вийде невдалою, сайт буде невдалим.

Як вибрати найкращу структуру для свого сайту

Основним принципом відмінної структури веб-сайту є інформаційна архітектура (IA). IA гарантує, що вміст організовано, структуровано та позначено ефективно та послідовно. Щоб створити найкращу інформаційну архітектуру для свого веб-сайту, потрібно враховувати наступні фактори:

Шлях користувача: оскільки веб-сайти створюються для обслуговування користувачів, важливо враховувати, як вони можуть відчувати ваш сайт або взаємодіяти з ним, а також їхні очікування щодо того, як він повинен працювати. Вміст: структура вашого веб-сайту також значною мірою визначатиметься типом та обсягом вмісту на сайті. Структура сайту електронної комерції буде відрізнятися від структури академічного сайту.

Контекст: Контекст веб-сайту визначається його бізнес-цілями, культурним контекстом, в якому він існує, і доступними ресурсами. Важливо враховувати цей факт, структуруючи веб-сайт.

Структура вашого сайту – це те, як різні сторінки сайту пов'язуються одна з одною за допомогою внутрішніх посилань та ієрархії. Це те, як інформація на сайті організована та представлена так, щоб алгоритм добре читав її контекст (рис. 1.1).

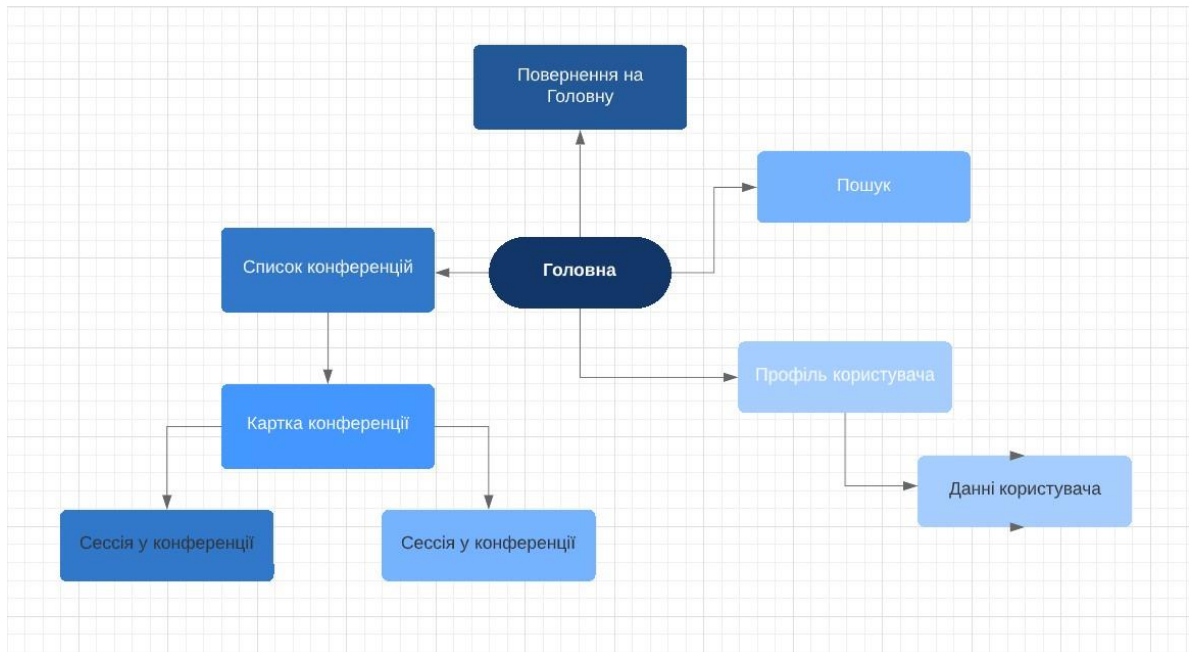


Рис. 1.1. Структура веб додатку

Загальна інформація:

Розмір сайту – 1450 пікселів.

Шапка і футер розтягуються по ширині екрана і однакові для всіх сторінок. Семейство шрифтів: Cambria, Century, Georgia. Можна вказати та інші популярні шрифти.

Розмір шрифтів (для Cambria):

- текст під логотипом в шапці – 16px;
- посилання в шапці – 14px; текст в футері – 16px.

Активні елементи:

Всі посилання виділені, прибираємо підкреслення при наведенні, кольори посилань на кілька темних відтінків (при перегляді виконавця). Колір кнопки, при наведенні з'являється курсор у вигляді руки.

Рекомендую робити окремі макети і описувати поведінку всіх посилань, кнопок, випадаючих меню, випадаючих вікон.

Під кожною категорією виводити 6 найпопулярніших за кількістю відгуків конференцій. Якщо в категорії є ще конференція, на неї немає посилання. Якщо більше конференцій немає, в категорію входить посилання «Показати все».

Дизайн змінюється таким чином, щоб всі зміни можна було внести глобально, без редагування кожної сторінки окремо: □ додано сірий фон контентного блоку;

- додано білий границю у картку;
- додано місце під блок над відгуками.

1.5.2 Прототип

Прототип. Самий наглядний і однозначний спосіб, щоб заздалегіть побачити як може виглядати сайт та зрозуміти, що треба змінити, а що залишити (рис. 1.2).

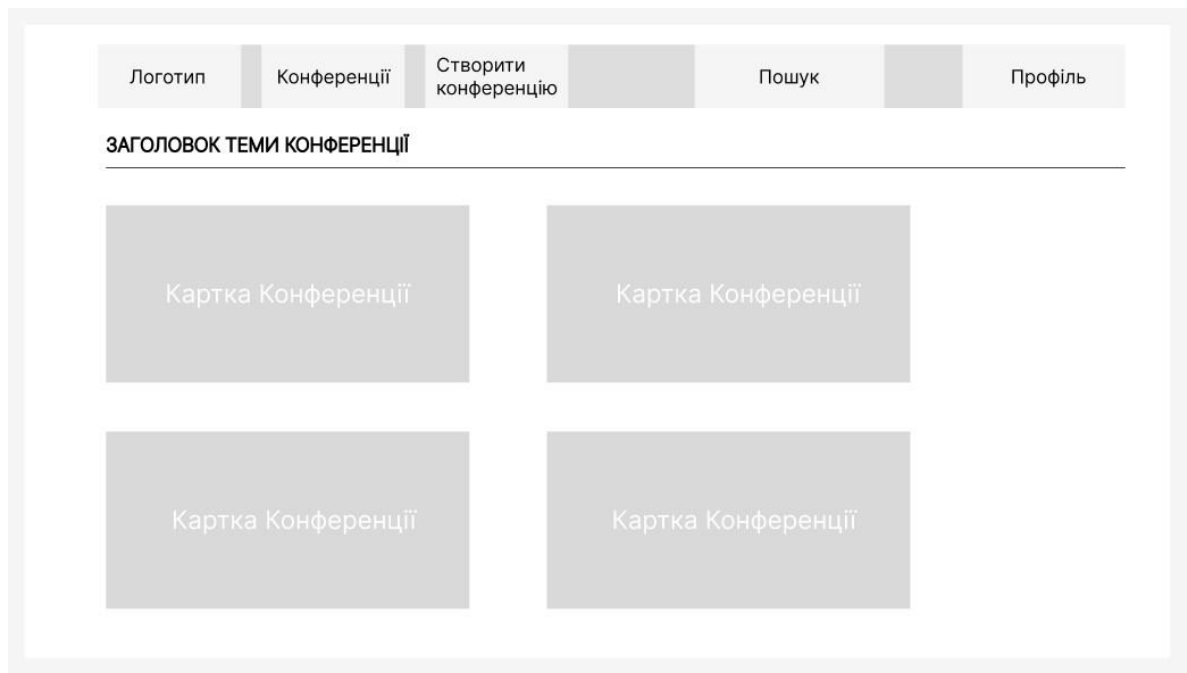


Рис. 1.2. Прототип головної сторінки веб додатку

РОЗДІЛ 2

ОПИС ПРОГРАМНИХ ЗАСОБІВ

2.1 Мова програмування JavaScript

JavaScript (JS) — це мова сценаріїв, яка в основному використовується у Вебі. Вона використовується для покращення сторінок HTML і зазвичай вбудований у HTML-код. JavaScript є інтерпретованою мовою. Таким чином, його не потрібно компілювати. JavaScript відтворює веб-сторінки в інтерактивній та динамічній формі.

Сторінки HTML чудово підходять для відображення статичного вмісту, наприклад просте зображення або текст. Однак сьогодні більшість сторінок рідко статичні. Багато сучасних сторінок мають меню, форми, слайд-шоу і навіть зображення, які забезпечують взаємодію з користувачем. JavaScript — це мова, яку використовують веб-розробники для забезпечення такої взаємодії. Оскільки JavaScript працює зі сторінками HTML, розробник повинен знати HTML, щоб використати весь потенціал цієї мови сценаріїв. Хоча існують інші мови, які можна використовувати для написання сценаріїв, на практиці це, по суті, весь JavaScript.

Існує два способи використання JavaScript у файлі HTML. Перший передбачає вбудовування всього коду JavaScript в код HTML, тоді як другий метод використовує окремий файл JavaScript, який викликається з елемента Script, тобто укладений тегами Script. Файли JavaScript визначаються розширенням .js. Хоча JavaScript в основному використовується для взаємодії з об'єктами HTML, його також можна зробити для взаємодії з іншими об'єктами, які не є HTML, такими як плагіни браузера, властивості CSS (каскадні таблиці стилів), поточна дата або сам браузер. Щоб написати код JavaScript, вам потрібен лише базовий текстовий редактор, наприклад Блокнот у Windows, Gimp в Linux або BBEdit. Деякі текстові редактори, як-от BBEdit, мають підсвічування синтаксису для JavaScript. Це

дозволить вам легко ідентифікувати елементи коду JavaScript. Останні версії Internet Explorer, Firefox і Opera підтримують JavaScript.

2.1.1 Як JavaScript робить речі динамічними

HTML визначає структуру вашого веб-документа та вміст у ньому. CSS оголошує різні стилі для вмісту веб-документа.

HTML і CSS часто називають мовами розмітки, а не мовами програмування, оскільки вони, по суті, забезпечують розмітку для документів з дуже невеликим динамізмом.

JavaScript, з іншого боку, є динамічною мовою програмування (рис. 2.1), яка підтримує математичні обчислення, дозволяє динамічно додавати вміст HTML до DOM, створює динамічні оголошення стилів, витягує вміст з іншого веб-сайту та багато іншого.

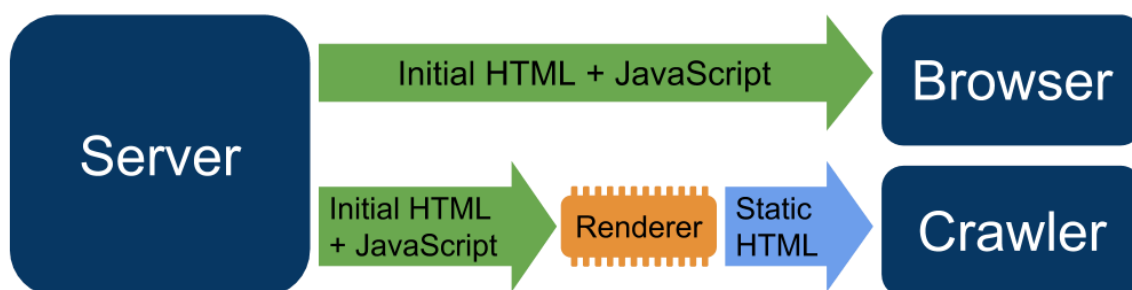


Рис. 2.1. Динамічний рендеринг веб-сторінок JavaScript

2.2 Мова програмування TypeScript

TypeScript — це надбудова над мовою JavaScript, який має єдиний компілятор з відкритим кодом (рис. 2.2) і розробляється в основному одним постачальником: Microsoft. Мета TypeScript — допомогти виявити помилки на ранній стадії за допомогою системи типів і зробити розробку JavaScript більш ефективною.



Рис. 2.2. Компіляція TypeScript в JavaScript

2.2.1 TypeScript досягає своїх цілей трьома способами:

- Підтримка сучасних функцій JavaScript. Мова JavaScript (а не середовище виконання) стандартизована за допомогою стандартів ECMAScript. Не всі браузери та середовища виконання JavaScript підтримують усі функції всіх стандартів ECMAScript (див. цей огляд). TypeScript дозволяє використовувати багато найновіших функцій ECMAScript і перекладає їх на старіші цільові елементи ECMAScript за вашим вибором (див. список цільових об'єктів компіляції під параметром компілятора `--target`). Це означає, що ви можете безпечно використовувати нові функції, такі як модулі, лямбда-функції, класи, оператор поширення та деструктуризацію, залишаючись зворотною сумісністю зі старими браузерами та середовищами виконання JavaScript.
- Розширена система типів. Підтримка типів не є частиною стандарту ECMAScript і, швидше за все, ніколи не буде пов'язана з інтерпретованою природою замість компільованого JavaScript. Система типів TypeScript неймовірно багата і включає в себе: інтерфейси, перерахування, гібридні типи, генерики, типи об'єднання/перетину, модифікатори доступу та багато іншого.

Офіційний веб-сайт TypeScript надає огляд цих функцій. Система типів Typescript є на одному рівні з більшістю інших типізованих мов і в деяких випадках, можливо, є більш потужною.

- Підтримка інструментів розробника – компілятор TypeScript може працювати як фоновий процес, щоб підтримувати як поетапну компіляцію, так і інтеграцію IDE, щоб ви могли легше орієнтуватися, виявляти проблеми, перевіряти можливості та реорганізувати свою кодову базу.

TypeScript підвищує продуктивність, допомагаючи уникнути багатьох помилок. Використовуючи типи, ми можемо виявити помилки під час компіляції замість того, щоб вони траплялися під час виконання.

2.3 Angular

Angular — це платформа та фреймворк для створення односторінкових клієнтських програм за допомогою HTML і TypeScript. Angular написаний на TypeScript. Він реалізує основні та додаткові функції у вигляді набору бібліотек TypeScript, які ви імпортуєте у свої програми.

Архітектура програми Angular спирається на певні фундаментальні концепції. Основними будівельними блоками фреймворка Angular є компоненти Angular, які організовані в NgModules. NgModules збирає пов'язаний код у функціональні набори; Angular додаток визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувати, і зазвичай має набагато більше модулів функцій.

Компоненти визначають перегляди, які представляють собою набори елементів екрану, які Angular може вибирати та змінювати відповідно до логіки та даних вашої програми.

Компоненти використовують служби, які надають конкретні функції, не пов'язані безпосередньо з представленнями. Постачальники послуг можуть бути введені в компоненти як залежності, що робить ваш код модульним, багаторазовим та ефективним.

Модулі, компоненти та служби — це класи, які використовують декоратори. Ці декоратори позначають свій тип і надають метадані, які повідомляють Angular, як їх використовувати.

Метадані для класу компонента пов'язують його з шаблоном, який визначає подання. Шаблон поєднує звичайний HTML з директивами Angular і розміткою прив'язки, які дозволяють Angular змінювати HTML перед відображенням його для відображення.

Метадані для класу служби надають інформацію, необхідну Angular, щоб зробити її доступною для компонентів за допомогою ін'єкції залежностей (DI).

Компоненти програми зазвичай визначають багато представлень, розташованих ієрархічно. Angular надає службу Router, щоб допомогти вам визначити шляхи навігації між представленнями. Маршрутизатор надає складні навігаційні можливості в браузері.

Як платформа, Angular включає:

- Компонентний фреймворк для створення масштабованих веб-додатків. □
Колекція добре інтегрованих бібліотек, які охоплюють широкий спектр функцій, включаючи маршрутизацію, керування формами, зв'язок клієнтсервер.
- Набір інструментів для розробників, які допоможуть вам розробляти, створювати, тестувати та оновлювати код.
- З Angular ви користуєтеся перевагами платформи, яка може масштабуватися від проектів окремого розробника до програм корпоративного рівня.
- Angular розроблено, щоб зробити оновлення максимально простим, тому скористайтеся перевагами останніх розробок з мінімумом зусиль.
- Найкраще те, що екосистема Angular складається з різноманітної групи з понад 1,7 мільйонів розробників, авторів бібліотек і творців контенту.

2.3.1 Основні відмінності між JavaScript і Angular

Існують різні основні відмінності між JavaScript та Angular. Деякі з них такі:

- JavaScript є як серверною, так і клієнтською мовою сценаріїв для створення веб-додатків. З іншого боку, Angular робить веб-додатки швидкими та простими з самого початку.
- JavaScript займає менше часу для виправлення помилок і дефектів у широкому масштабі. У порівнянні з JavaScript, Angular потребує більше часу, щоб зробити те ж саме.
- JavaScript є одним з найефективніших методів веб-розробки, що використовуються для створення веб-додатків. З іншого боку, Angular здебільшого використовувався як фреймворк JS для розробки веб-додатків.
- JavaScript - це мова програмування, яка використовується для маніпулювання DOM (об'єктна модель документа). З іншого боку, Angular розширює свої можливості за допомогою різних технологій.
- JavaScript не підтримує впровадження залежностей. Тоді як Angular підтримує як зв'язування даних, так і ін'єкцію залежностей.
- Код JavaScript швидкий і швидкий. З іншого боку, програма Angular зазвичай стає повільною.
- JavaScript використовує ту саму стратегію протягом кількох років. З іншого боку, Angular було вдосконалено до машинопису, що робить програми легшими та інтерактивнішими.
- JavaScript має розширений інтерфейс користувача, який включає повзунки та інші функції. З іншого боку, Angular — це керована даними платформа, яка використовується для створення веб-додатків.
- JavaScript — потужна і складна мова програмування. З іншого боку, Angular є простим і ефективним фреймворком.

Обидві ці веб-технології використовуються для створення веб-додатків. JavaScript - це мова програмування з відкритим вихідним кодом і об'єктноорієнтована мова, а Angular - це фреймворк з відкритим вихідним кодом, заснований на моделі MVC. JavaScript — це мова програмування для створення веб-сайтів, що покращує інтерактивність веб-сайту. Це допомагає маніпулювати вмістом на веб-сайтах, щоб перевірити відгуки користувачів у браузері, таким

чином впливаючи на дії користувача, включаючи динамічний вміст, як-от елементи перетягування, повзунки та багато інших. Це одна з трьох основних технологій всесвітньої павутини та основа всіх технологій JavaScript.

З іншого боку, Angular — це фреймворк з відкритим кодом, який спеціалізується на розробці масивних односторінкових додатків (рис. 2.3) і написаний на JavaScript. Він заснований на архітектурі MVC. Це підхід, керований даними, який дозволяє веб-додаткам мати більшу бібліотеку HTML. Це платформа для майбутнього, щоб задовольнити великі вимоги до даних без необхідності оновлювати моделі.

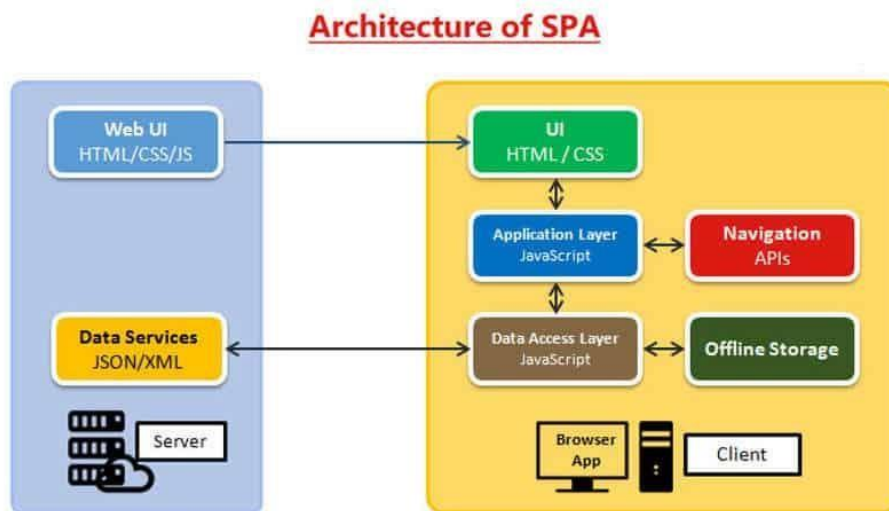


Рис. 2.3. Односторінкова архітектура

2.4 RxJs

Повною формою RxJS є Reactive Extension для Javascript. Це бібліотека javascript, яка використовує спостережувані елементи для роботи з реактивним програмуванням, яке має справу з асинхронними викликами даних, зворотними викликами та програмами на основі подій. RxJS можна використовувати з іншими бібліотеками та фреймворками Javascript. Він підтримується javascript, а також машинописом.

RxJS визначається як бібліотека для складання асинхронних програм і програм на основі подій за допомогою спостережуваних послідовностей. Він надає один основний тип, Observable, типи супутників (Observer, Schedulers, Subjects) та оператори, створені за допомогою Array#extras (map, filter, reduce, every тощо), щоб дозволити обробляти асинхронні події як колекції.

Бібліотека RxJS чудово підходить для обробки асинхронних завдань. Він має велику колекцію операторів для фільтрації, обробки помилок, умов, створення, багатоадресної передачі тощо. Він підтримується JavaScript і TypeScript, і він добре працює з Angular.

Асинхронність потрібна щоб головний "процес" програми не блокував виконання інструкцій. А передавав управління іншому незалежному потоку/процесу, доки минулий не відповість успіхом про успішність завершення виконання та готовність далі йти за кодом/інструкціями.

Асинхронне програмування JavaScript не пов'язане з багатопоточністю. JavaScript – однопоточковий, це означає, що не існує стандартних мовних конструкцій, які дозволять створити у додатку додатковий потік для виконання паралельних обчислень (єдина можливість організувати справжню багатопоточність – використовувати Web Workers). Асинхронне програмування – стиль програмування, у якому результат роботи функцій доступний відразу, а через деякий час. Асинхронна функція – це функція, після виклику якої JavaScript програма продовжує працювати, тому що функція одразу виконує повернення. Результат роботи асинхронної функції стає відомим пізніше, і для того, щоб сповістити нашу програму про отримані значення, асинхронна функція викликає іншу функцію (callback), яку ми передаємо в аргументах при запуску.

Використання callback функцій може призвести до появи проблеми, яку називають Pyramid of Doom - callback функція, в якій викликається асинхронна функція, якою передається callback функція, і в ній викликається асинхронна функція.

Ми хочемо спостерігати за даними, а це означає, що існує певний виробник даних, який може бути сервером, який надсилає дані за допомогою HTTP або

WebSockets, поля введення інтерфейсу користувача, куди користувач вводить деякі дані. Спостережуваний — це функція на клієнті, яка отримує дані виробника та передає їх передплатнику. Спостерігач — це об'єкт, який знає, як обробляти елементи даних, які висуваються об'єктом спостереження (рис. 2.4).

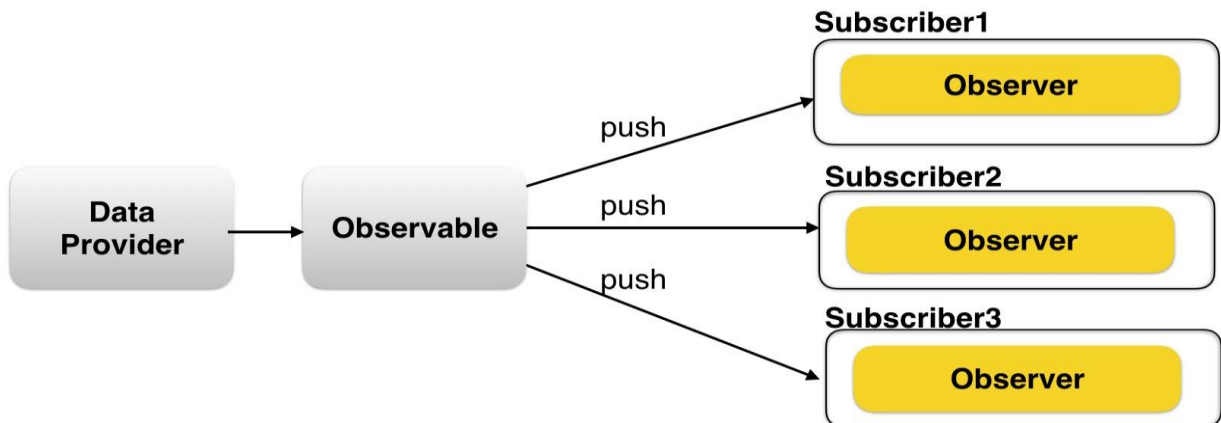


Рис. 2.4. Спостерігач в RxJs

2.5 Figma

Figma — це хмарний інструмент для проектування та створення прототипів для цифрових проектів. Він створений для того, щоб користувачі могли співпрацювати над проектами та працювати практично будь-де.

За допомогою Figma можна вносити зміни у проєкт в реальному часі через веб-браузер. Хоча цей інструмент дуже схожий на інші варіанти створення прототипів, ключовою відмінністю є можливість працювати з командами над проєктами. На сьогоднішній день Figma є, мабуть, найбільш кращим варіантом.

Figma — це векторний інструмент, який живе в хмарі, що дозволяє користувачам працювати в будь-якому місці з браузера. Це швидкий інструмент, який створений для проектування, створення прототипів, співпраці та організаційного проектування систем.

Як і багато інших інструментів, Figma підтримується потужною спільнотою дизайнерів і розробників, які спільно використовують плагіни для збільшення

функціональності та прискорення робочих процесів. Будь-хто може зробити свій внесок і поділитися.

Figma використовується деякими великими брендами, включаючи Slack, Twitter, Zoom, Dropbox і Walgreens. Ці назви свідчать про те, що цей інструмент достатньо надійний, щоб забезпечити роботу практично будь-якого проекту.

Однією з головних характеристик цього інструменту є його легка доступність. Існує веб-версія, доступна через браузер, а також настільна версія для Windows та MacOS.

Ще однією спільною функцією, яку пропонує Figma, є експорт коду. Це дозволяє дизайнерам ділитися CSS-кодом веб-макетів із розробниками інтерфейсів, щоб дозволити їм перевірити кожен елемент макету. Також є можливість експортувати код iOS та Android у мобільні додатки для кращого потоку.

Основні функції включають:

- Сучасний інструмент «Перо», який дозволяє малювати в будь-якому напрямку та миттєво створювати дуги.
- Відкрийте функції шрифту Type.
- Автоматизовані завдання за допомогою плагінів для повторюваних елементів для прискорення проектів.
- Інтелектуальний інструмент вибору з автоматичним налаштуванням інтервалів, упорядкування й упорядкування.
- Гнучкі стилі, які зберігають прямо у вашому проекті.
- Створення проектних систем і компонентів.
- Перетягніть доступні бібліотеки.
- Можливість мати необмежену кількість глядачів на проекті, щоб вся команда була в курсі.
- Перевірте файли дизайну та візьміть фрагменти коду.
- Проста функція експорту, яка має пряме посилання.
- Дизайн взаємодії та оптимізовані для мобільних пристроїв прототипи.
- Розумна анімація для з'єднання об'єктів і переходів.

- Вбудовані коментарі.
- Редагуйте разом з командами в режимі реального часу в спільних дизайнерських просторах.
- Історія версій, яка дає вам знати все, що змінилося, і хто це зробив.
- Створюйте системи дизайну з ресурсами, доступними для пошуку, стилями для спільного використання та всіма всіма в одному місці.
- Створюйте багаторазові компоненти з можливістю замінити їх, коли це необхідно.
- Одна платформа для всього, від дизайну до прототипу.

Отже, з Figma ми можемо працювати де завгодно, з Інтернетом або без нього. Це величезна перевага над конкурентами.

Додавання анімації дозволяє створювати більш реалістичні прототипи, які занурюють нас в UX. Отже, це дозволяє отримати більш точні та реалістичні відгуки, оскільки вони будуть на основі інтерактивних прототипів.

2.6 Протоколи взаємодії HTTP\HTTPS

HTTP був створений на основі пропозиції побудувати гіпертекстову систему для Інтернету. Це було ще до того, як це було передано у всесвітню мережу. Протягом перших днів 1990 року HTTP був побудований на основі існуючих протоколів TCP та IP для обміну документами HTML.

З тих пір багато чого змінилося, і через його простоту, яка колись була вигідною для стандартизації, HTTP вже на виході. Оскільки обмін даними через HTTP проходить як звичайний текст, можливість перехопити ці прості дані стала занадто легкою для кіберзлочинців. Оскільки HTTP надає інформацію на вимогу, але не визначає пріоритети надходження інформації, одержувач стає вразливим. HTTPS — це остання версія HTTP, яка базується на його функціональних можливостях, щоб забезпечити безпеку його можливостей. В основному вони однакові. Проте HTTPS пропонує користувачам спокій у захисті передачі даних.

Технологія HTTPS працює шляхом простого шифрування даних користувача, коли вони переміщуються між їхнім веб-браузером і сервером веб-сайту. Шкодюючи ці дані, будь-хто, хто намагається перехопити, не зможе їх прочитати, захищаючи таким чином від хакерів і ботів. Сторінки використовуватимуть TLS (рівень безпеки транспорту) або SSL (рівень безпечних сокетів). TLS є новішою версією стандартного SSL і в основному є одним і тим же. Коли запитується безпечно з'єднання через HTTPS, SSL / TLS використовує відкритий ключ для шифрування даних і закритий ключ для розшифрування інформації після отримання. Це називається рукостисканням SSL, коли обидві сторони підтверджують авторизацію через захищене з'єднання за допомогою секретного ключа.

Переваги переходу старого веб-сайту з HTTP на HTTPS або створення нового веб-сайту з HTTPS неможливо переоцінити. Захист даних користувачів від хакерів має бути пріоритетом для кожного веб-сайту, незалежно від того, чи це блог, сайт місцевого бізнесу чи повноцінний сайт електронної комерції. Одне порушення може легко підірвати репутацію малого бізнесу та його майбутнє зростання. Завдяки впевненості клієнтів у заходах безпеки бізнесу, HTTPS забезпечує перевагу від конкурентів, які не перейшли з HTTP (рис 2.5).

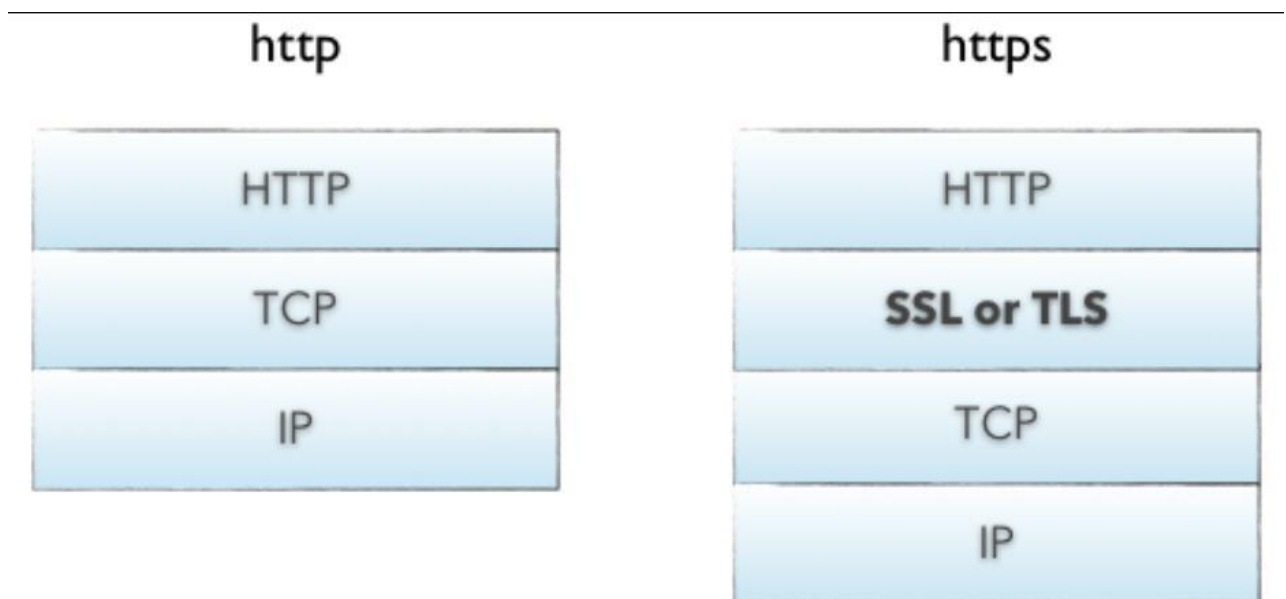


Рис. 2.5. Різниця HTTP\HTTPS

Коротка версія така: шифрування даних стає стандартною нормою в Інтернеті, і щоб створити веб-сайт з довірою та впевненістю, HTTPS є обов'язковим. Незалежно від того, наскільки великим чи малим буде онлайн-бізнес чи інший веб-сайт, HTTPS пропонує користувачам золотий стандарт безпеки, а для ваших потреб є сертифікат безпеки.

2.7 Rest API

RESTful API — це архітектурний стиль інтерфейсу прикладної програми, який використовує HTTP-запити для доступу та використання даних (рис. 2.6). Ці дані можна використовувати для типів даних GET, PUT, POST і DELETE, що стосується читання, оновлення, створення та видалення операцій, що стосуються ресурсів.

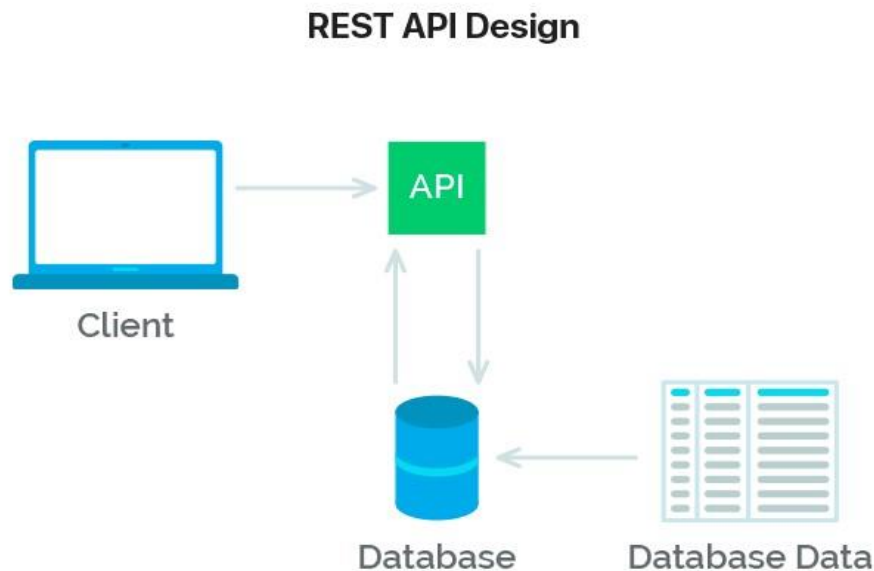


Рис. 2.6. Rest API архітектура

REST API – це інтерфейс програмного забезпечення, який може використовуватися кількома клієнтами для зв'язку з сервером. Rest API — це свого роду веб-сервіс, який зберігає та отримує необхідні дані. Він забезпечує

велику гнучкість розробникам, оскільки не потребує жодних залежних бібліотек коду для доступу до веб-сервісів.

RESTful API використовує команди для отримання ресурсів. Стан ресурсу в будь-який момент часу називається представленням ресурсу. RESTful API використовує існуючі методології HTTP, визначені протоколом RFC 2616, наприклад (рис. 2.7):

- GET, щоб отримати ресурс;
- PUT, щоб змінити стан або оновити ресурс, яким може бути об'єкт, файл або блок;
- POST, щоб створити цей ресурс;
- DELETE, щоб видалити його.

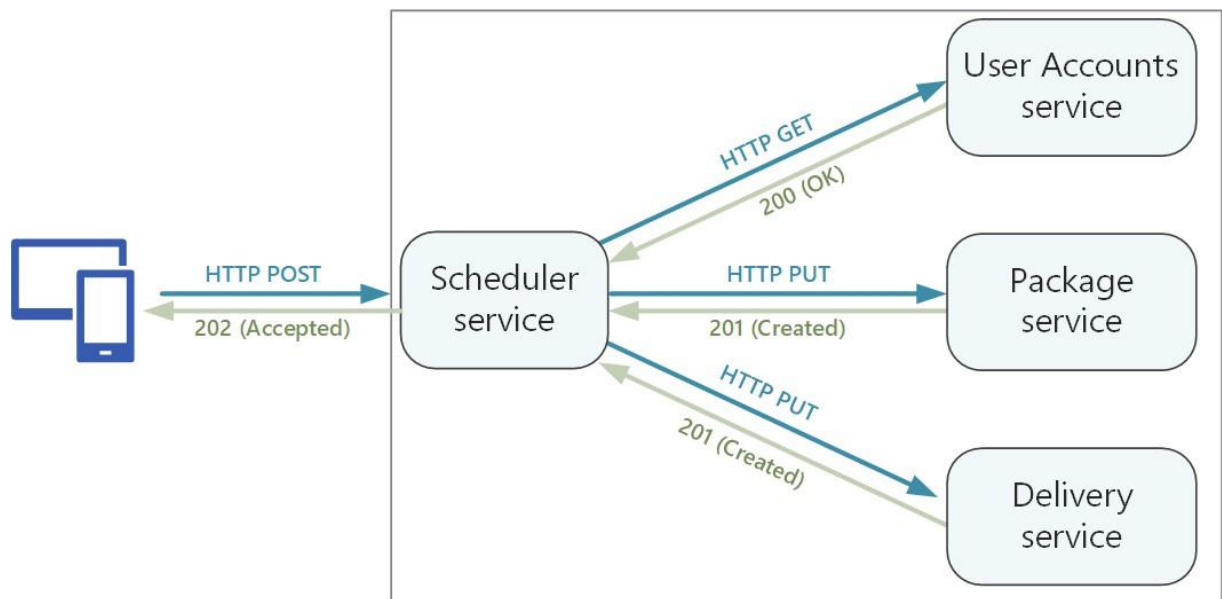


Рис. 2.7. HTTP запити

З REST мережеві компоненти є ресурсом, до якого користувач запитує доступ, як чорний ящик, деталі реалізації якого неясні. Усі дзвінки без громадянства.

Служба RESTful нічого не може зберегти між виконаннями.

2.8. Переваги засобів автоматизації тестування над ручним

2.8.1 Що таке автоматичне тестування?

Перетворення процесу ручного тестування як автоматичного називається автоматизованим тестуванням, яке включає розробку та виконання тестових сценаріїв для перевірки вимог клієнтів за допомогою будь-якого інструменту автоматизації.

Автоматичне тестування не є заміною ручного тестування. Ідея автоматизації тестування полягає в автоматизації деяких частин процесу тестування програмного забезпечення, але не всього процесу тестування.

2.8.2 Переваги автоматизованого тестування

Автоматизація процесу тестування значно швидше, ніж ручне тестування, оскільки воно займає більше часу.

Усуває людську помилку під час виконання одних і тих самих операцій кожного разу.

Ми можемо перевірити та побачити, як програма реагує після багаторазового виконання тих самих операцій.

Ми можемо створити набір тестів, який охоплює кожну функцію програми. За потреби ми можемо повторно використовувати ті самі тестові сценарії для різних версій програми.

Автоматичне тестування проти ручного тестування.

Головне — вирішити, коли використовувати автоматичне тестування, а коли — ручне тестування. Коли у нас буде більше роботи з регресійного тестування, ми можемо перейти до автоматизованого тестування, оскільки воно зменшує кількість повторюваних операцій. Коли ми тестуємо один-два рази або автоматизація тестування дорога, ми можемо перейти на ручне тестування (рис. 2.8).

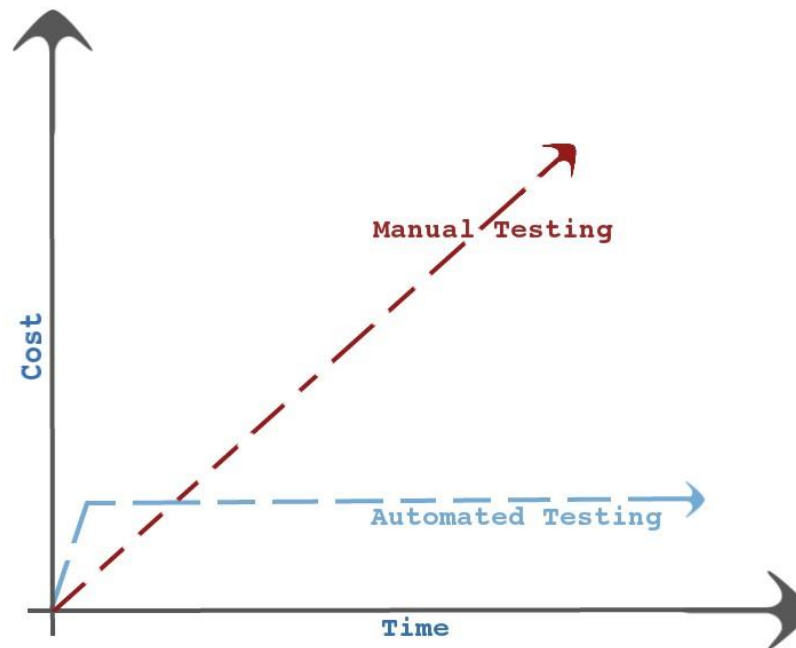


Рис. 2.8. Діаграма цін тестування

Плюси автоматизації

- Якщо нам доводиться виконувати набір тестів неодноразово, то автоматизація є правильним для нас.
- Автоматизовані тести можна запускати одночасно на різних машинах з різними платформами ОС.
- Автоматичне тестування збільшує швидкість тестування, якість і точність.
- Виконання одних і тих самих дій з різними входами набагато легше і швидше.

Мінуси автоматизації

- Спочатку автоматизація коштує дорожче.
- Неможливо автоматизувати візуальні посилання, як-от колір шрифту, розмір шрифту.
- Якщо інтерфейси користувача значно змінюються, знову потрібно виконати обслуговування сценаріїв.

Плюси посібника

- Дозволяє тестувальнику виконувати більше спеціальних тестів (випадкове тестування).
- Покращує комунікацію між командою тестування, командою розробників і взаємодією з клієнтом.

Мінуси керівництва

- Ручне тестування займає багато часу і вимагає інвестицій у людські ресурси.
- Тестування програми на різних машинах з різними конфігураціями відбувається послідовно.

2.8.3 Види засобів автоматизації тестування

Інструменти функціонального тестування.

- Інструменти тестування продуктивності.
- Інструменти управління тестуванням.
- Інструменти звітування про помилки.

Критерії вибору інструменту

Той, хто збирається використовувати інструмент автоматизації, врахуйте ці міркування, перш ніж вибрати:

- Вартість ліцензування.
- Функції, які пропонує інструмент.
- Технології, які підтримує інструмент.
- Середовище/платформи, які підтримує інструмент.
- Зручність у використанні.
- Продуктивність.
- Підтримка постачальників.

2.8.4 Життєвий цикл автоматизації тестування

Життєвий цикл автоматизованого тестування включає багато етапів/кроків.

Основними етапами життєвого циклу є:

- Планування тестів.
- Проаналізуйте програму, що тестується.
- Налаштувати тестове середовище.
- Розробка тестових сценаріїв.
- Поліпшити тестові сценарії.
- Налагодити тести.
- Виконати тести.
- Проаналізуйте результати тестування.
- Дефекти/Звіти.

Висновок

Інструмент автоматизації економить час, ресурси та дає точні результати.

2.9 Karma

Angular надає надійні інструменти тестування з коробки. Коли ви створюєте проект Angular за допомогою інтерфейсу командного рядка, він поставляється з повністю робочою настройкою тестування для модульних, інтеграційних та наскрізних тестів.

Karma — це прямий продукт команди Angular, яка намагається перевірити власні функції фреймворку за допомогою існуючих інструментів. В результаті цього вони створили Karma і перевели її на Angular як тестовий запуск за замовчуванням для програм, створених за допомогою Angular CLI.

Karma, по суті, є інструментом, який створює веб-сервер, який виконує вихідний код проти тестового коду для кожного з підключених браузерів. Результати кожного тесту для кожного браузера перевіряються та відображаються через командний рядок розробнику, щоб він міг побачити, які браузери та тести пройшли чи не пройшли.

Модульне тестування (Unit Testing) – це тип тестування програмного забезпечення, у якому тестуються окремі модулі чи компоненти програмного забезпечення. Його ціль полягає в тому, щоб перевірити, що кожна одиниця програмного коду працює належним чином (рис. 2.9).

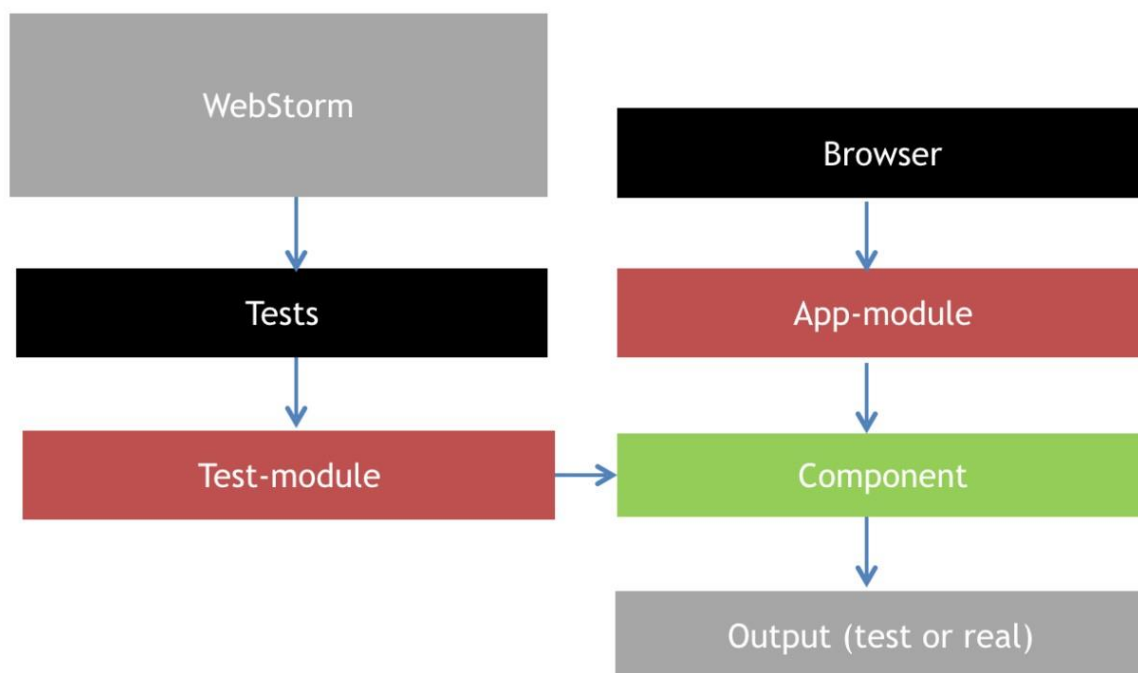


Рис. 2.9. Модульне тестування

Як ви бачили, модульні тести — це тести, які ви пишете, щоб перевірити, чи дуже маленькі частини програми поведуться належним чином. Ці маленькі деталі називаються одиницями, звідси й назва «юнітне тестування». Під час модульного тестування ви перевіряєте, чи блоки працюють правильно, незалежним як від інших блоків, так і від зовнішніх проблем.

Такі обмеження дають цікаві та цінні результати. Для початку, модульні тести, як правило, швидкі, оскільки вони не взаємодіють з речами, які можуть їх уповільнити. Крім того, модульні тести дають дуже точний зворотний зв'язок; оскільки вони не покладаються ні на що інше, окрім тестованого коду, вони не

можуть вийти з ладу з випадкових причин, наприклад, проблема автентифікації бази даних або відключення мережі.

І останнє, але не менш важливе, модульні тести є детермінованими. Якщо модульний тест наразі проходить, він продовжуватиме проходити, якщо хтось не внесе зміни в виробничий код — або сам тестовий код — що призведе до його невдачі. Вірно також і навпаки: якщо тест провалився, він не почне проходити раптово. Ця властивість означає, що модульні тести не завершуються випадково, що допомагає розробникам отримати більше довіри до них.

Інструменти тестування, які постачаються з Angular, є низькорівневими. Вони лише забезпечують основні операції. Якщо використовувати ці інструменти безпосередньо, тести стають безладними, повторюваними і важкодоступними. Тому нам слід створити інструменти високого рівня тестування, які втілюють умови в код, щоб писати короткі, читабельні та зрозумілі тести.

Ідея Karma полягає в тому, щоб виконувати ті самі тести в різних браузерах, щоб забезпечити взаємодію між браузерами.

Цикл зворотного зв'язку дозволяє працювати над впровадженням та тестувати код пліч-о-пліч. Це важливо для тестування розробки. Ми змінюємо реалізацію і очікуємо, що тест провалиться – тест «червоний». Ми адаптуємо тест, щоб він знову пройшов – тест «зелений». Або спочатку пишемо невдалий тест, а потім адаптуємо реалізацію, поки тест не пройде.

Звичайно, ручне тестування все ще є цінним у багатьох ситуаціях. Однак практично неможливо постачати високоякісне програмне забезпечення зі швидкістю, якої вимагає сучасний світ, покладаючись лише на ручні процеси. Вони просто не можуть масштабуватися.

РОЗДІЛ 3

ЗАГАЛЬНІ ВІДОМОСТІ ДОДАТКУ

3.1 Розробка концепції

Аналізуючи актуальність додатку, я вирішила, що розробити веб версію.

Цільова аудиторія – студенти, котрим потрібно здобути більше знань; працівники, яким потрібно підвищити свою кваліфікацію; школярі, котрі тільки починають свій шлях вибору професії; люди, які хочуть змінити діяльність (рис. 3.1).



Рис. 3.1. Цільова аудиторія

Головна ідея цього додатку підвищити продуктивність у навчанні, відстежуючи навчальні конференції, обмінюватись знаннями та розширити своє коло знайомих у професійній сфері.

Короткий опис застосування – це веб додаток націлений на відстеження навчальних семінарів. Ви можете створювати нову конференцію, заповнивши поля з описом події. У кожній конференції є сесії, які можна відфільтрувати за рівнем складності та рейтингом.

Головні вимоги додатку:

- головна сторінка;
- список конференцій;
- створення нової конференції з валідацією;
- пошук конференцій з існуючих;
- профіль користувача з можливістю змінити його;
- можливість зайти у конференцію; • можливість відфільтрувати існуючі сесії;
- створення нової сесії.

Унікальність додатку:

- зручний та дружній інтерфейс;
- поєднання різних можливостей для відстеження конференцій;
- єдиний додаток для відстеження навчальних конференцій.

3.2 Проектування структури та створення додатку

Для розробки проекту веб додатку, я проаналізувала декілька інструментів, як наслідок було вирішено обрати Figma, що є зручним інструментом для проектування графічних інтерфейсів.

Головні елементи інтерфейсу додатку:

1. Header веб додатку.
2. Кнопка для переходу на головну сторінку.
3. Кнопка для відображення всіх конференцій.
4. Кнопка для створення нової конференції.
5. Поле для пошуку по сайту.

6. Кнопка профілю.
7. Картка з інформацією про конференцію.
8. Форма для створення конференції.
9. Кнопка для збереження форми.
10. Кнопка для відмови інформації з форми.
11. Поле форми.
12. Блок кнопок з фільтрами.
13. Кнопка фільтрації по рівню.
14. Кнопка фільтрації по рейтингу.
15. Кнопка створення нової сесії.
16. Поле профілю ім'я.
17. Поле профілю з фамілією.
18. Кнопка збереження.
19. Кнопка відмови.
20. Сторінка з сесіями.
21. Сторінка профілю.

Визначившись із головними елементами інтерфейсу можна починати працювати у Figma. Для початку я створила новий файл дизайну.

На головній сторінці має бути відображено всі наявні конференції які у нас заплановані. Повинна бути шапка сату з навігацією по додатку. (рис 3.2).

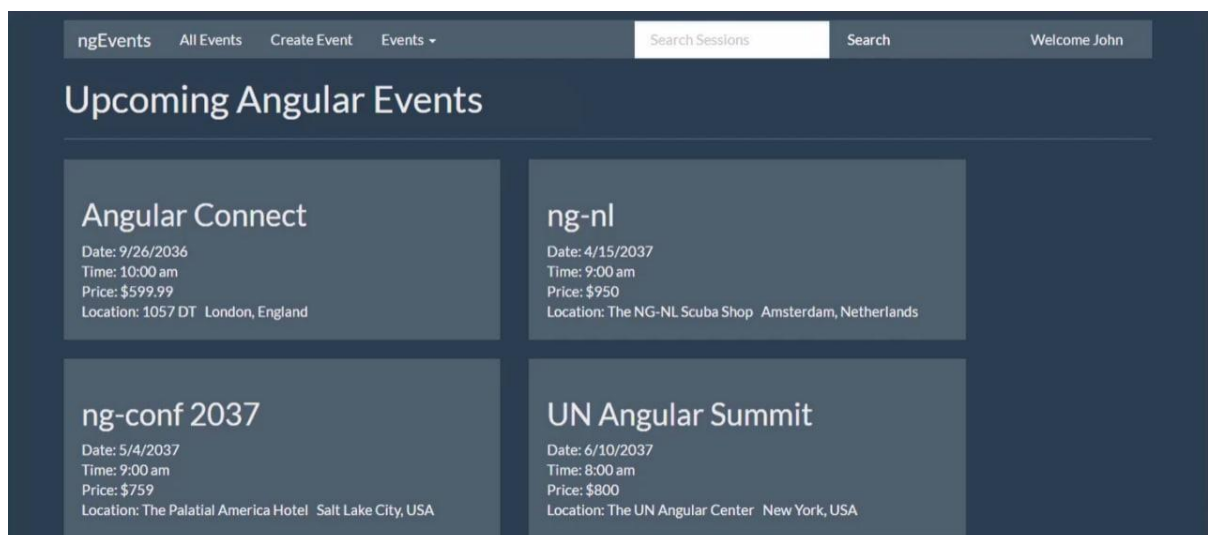


Рис. 3.2. Прототип початкової сторінки додатку

У шапці сайту повина бути навігація. Кнопка з переходом на головну сторінку, кнопка з перегляду всіх конференцій та кнопка для створення нової конференції (рис 3.3).

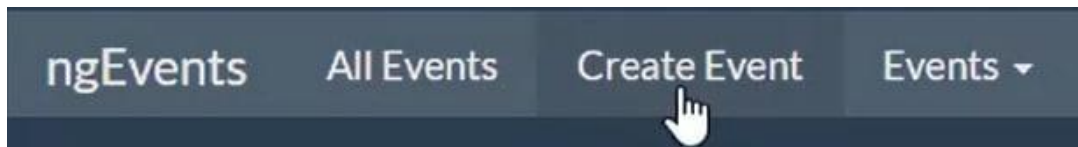


Рис. 3.3. Навігація у шапці веб додатку

Також у шапці має бути поле та кнопка для пошуку конференцій з вже існуючих на сайті (рис 3.4).



Рис. 3.4. Поле для пошуку

Останнє, що має бути у шапці сайту - це кнопка профілю користувача (рис 3.5).



Рис. 3.5. Кнопка профілю

Всі заплановані конференції мають відображатися на головній сторінці сайту у вигляді однакових карток (рис 3.6).

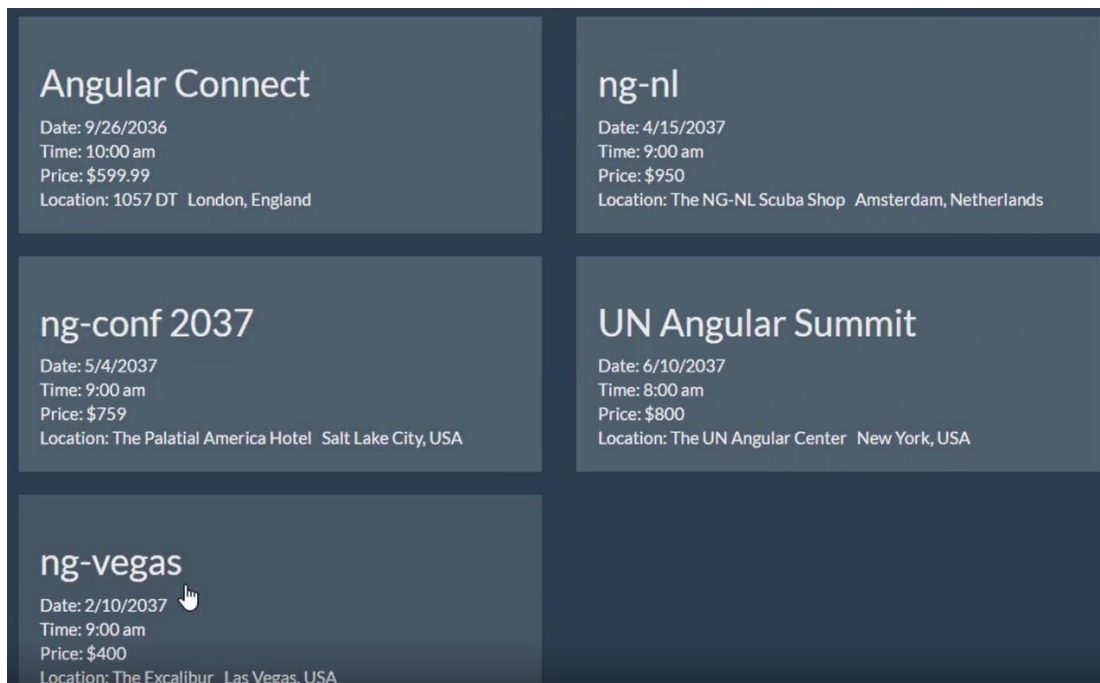


Рис. 3.6. Картки конференцій

Повинна бути можливість натиснути на картку з конференцією (рис 3.7) та перейти у опис цієї конференції (рис 3.8).

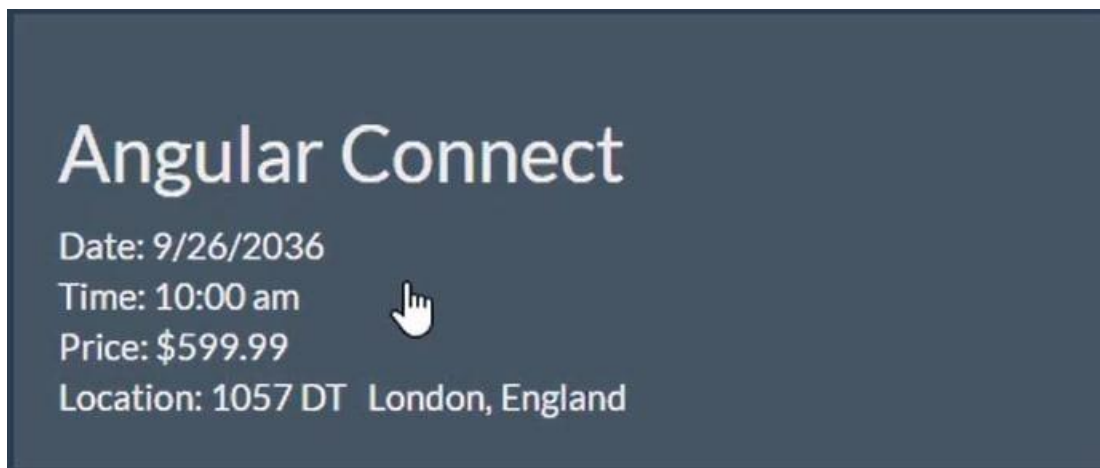


Рис. 3.7. Картка конференції

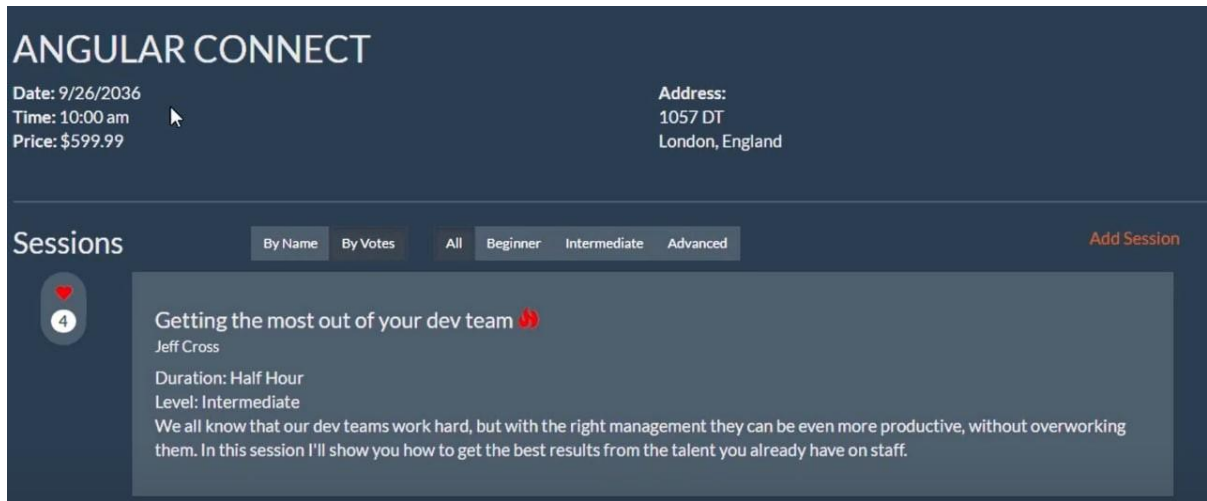


Рис. 3.8. Картка опису конференції

Також на стрінці з сесіями повинен бути блок з фільтрацією існуючих сесій (рис. 3.9).

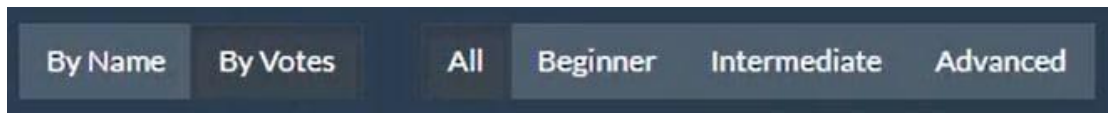


Рис. 3.9. Блок фільтрацій

Фільтрації повинні бути по рейтингу на іменам (рис. 3.10), а також по рівню складності (рис. 3.11).

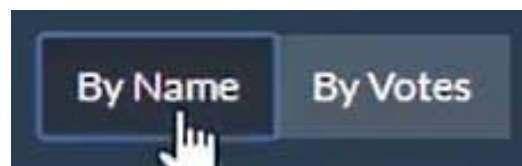


Рис. 3.10. Кнопки фільтрації по іменам та рейтингу

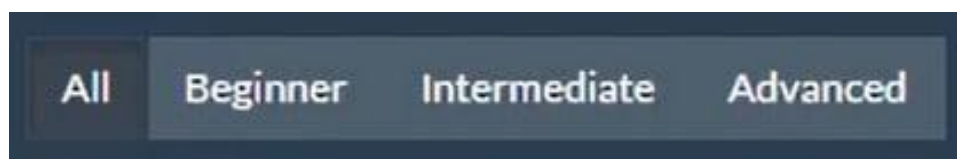


Рис. 3.11. Кнопки фільтрації по рівню складності

Також повинна бути кнопка з можливістю зайти у профіль користувача (рис. 3.12) та змінити дані за необхідності (рис. 3.13).



Рис. 3.12. Кнопки профілю



Рис. 3.13. Форма для змінення даних у профайлі

Кнопка повернення на головну сторінку веб додатку (рис. 3.14).



Рис. 3.14. Кнопка повернення на головну сторінку

У меню з навігацією - кнопка створення нової конференції (рис. 3.15) з формою та валідацією в відповідних полях (рис. 3.16).

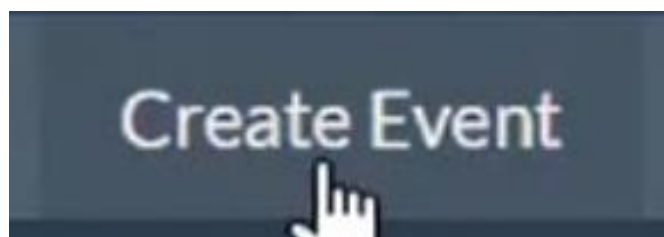


Рис. 3.15. Кнопка створення нової конференції

New Event

Event Name:
Name of your event...

Event Date:
format (mm/dd/yyyy)...

Event Time:
start and end time...

Event Price:
event price...

Event Location:
Address of event...

City... Country...

Рис. 3.16. Форма для створення конференції

Таким чином, створивши повноцінний прототип, за допомогою якого буде набагато швидше реалізовувати інтерфейс створюваного веб додатку, та прослідкувати зв'язки між сторінками.

Після створення прототипу можна розпочати розробку додатку з встановлення обраного середовища розробки WebStorm з офіційного сайту Jet Brains.

Після завантаження програми, потрібно створити новий проект.

Треба обрати Angular CLI (рис. 3.17) тому, що це дасть нам змогу дуже легко розробити додатки та розширювати його функціонал у майбутньому.

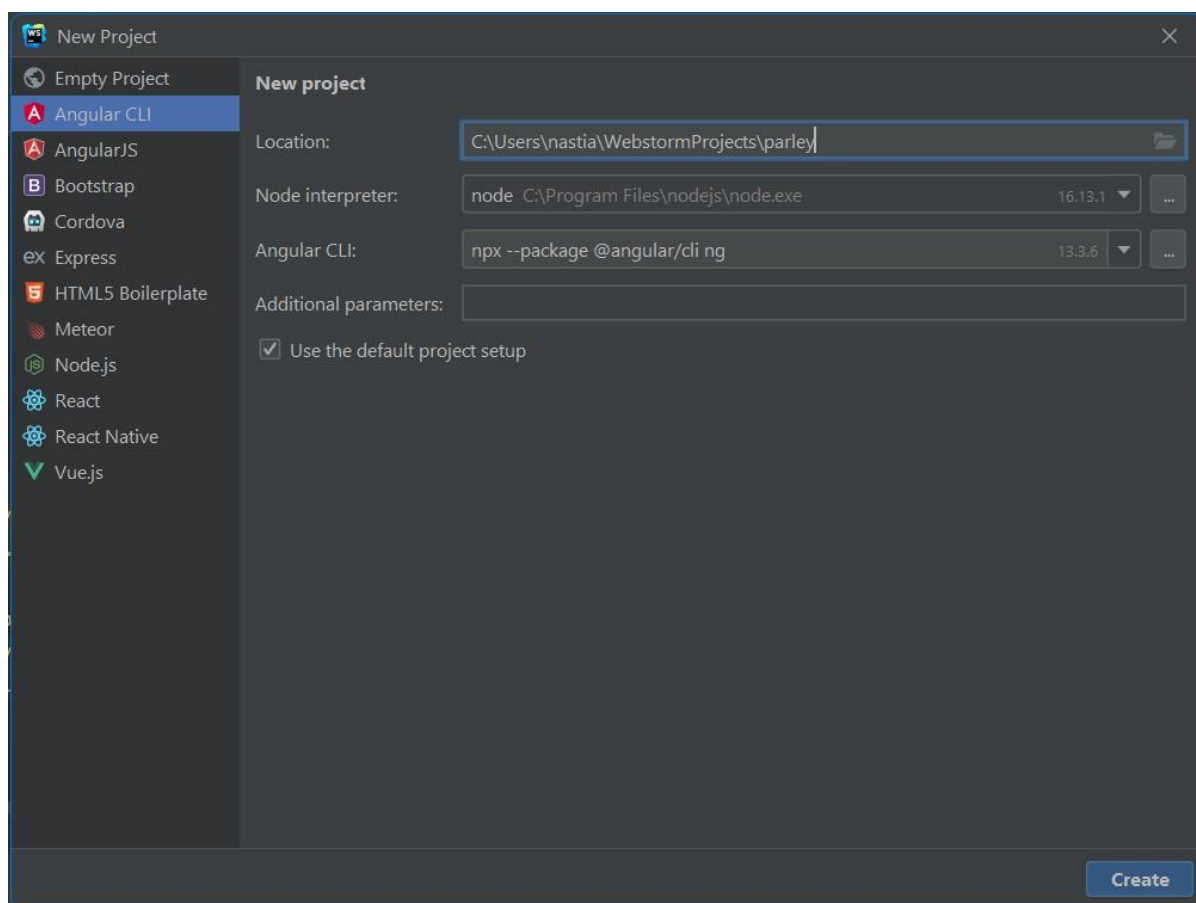


Рис. 3.17. Шаблон для створення веб додатку

Також нам не потрібно буде витратити час на налаштування всіх необхідних залежностей та з'єднання всіх разом.

Далі починаємо роботу з Git. Розгалуження дозволяє вийти з вихідної бази коду та ізолювати свою роботу від інших правок. Це також допомагає Git легко об'єднувати версії пізніше.

Git буде відстежувати зміни, які внесені у файл, також буде запис про те що вже зроблено, можна буде повертатися до версій, за необхідності. Також це спростить співпрацю, дозволяючи об'єднувати внесені зміни в одне джерело.

Щоб у майбутньому виробництві можна було розширювати додаток було вирішено обрати архітектуру GitFlow (рис. 3.18). Основна причина полягає в тому, що ми зможемо постійно підтримувати попередні версії у виробництві під час розробки наступної версії.

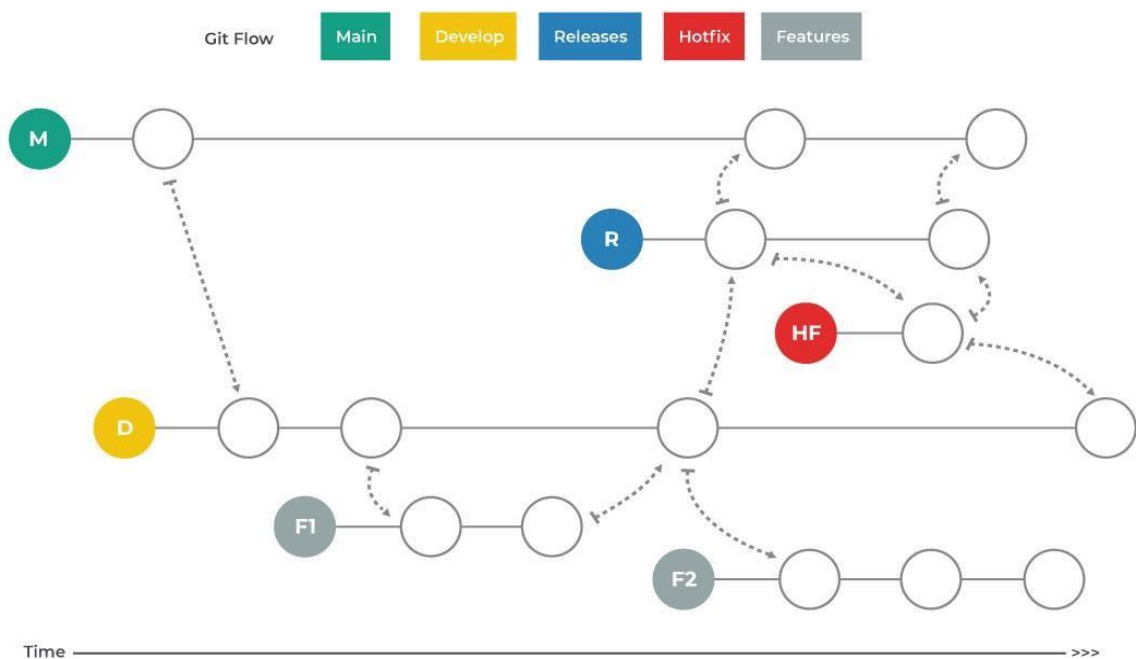


Рис. 3.18. Git Flow архітектура

Далі у профілі GitHub, створюємо новий репозиторій, та копіюємо посилання на його клонування. Повертаємося до коду, після чого вставляємо посилання на наш репозиторій, та публікуємо у GitHub.

Переходимо до GitHub, та бачимо, що наш проект вдало завантажився, далі ми зможемо відслідковувати його зміни.

Можемо починати роботу з ангуляр. Спочатку треба створити перший компонент із зовнішнім шаблоном та прив'язкою даних. Також створюємо CSS файли до кожного компонента. Оскільки Angular забезпечує вбудовану інкапсуляцію стилів (рис. 3.19), в нас не буде проблеми з витіканням стилів в інші компоненти, і не потрібно вигадувати назву стилю.

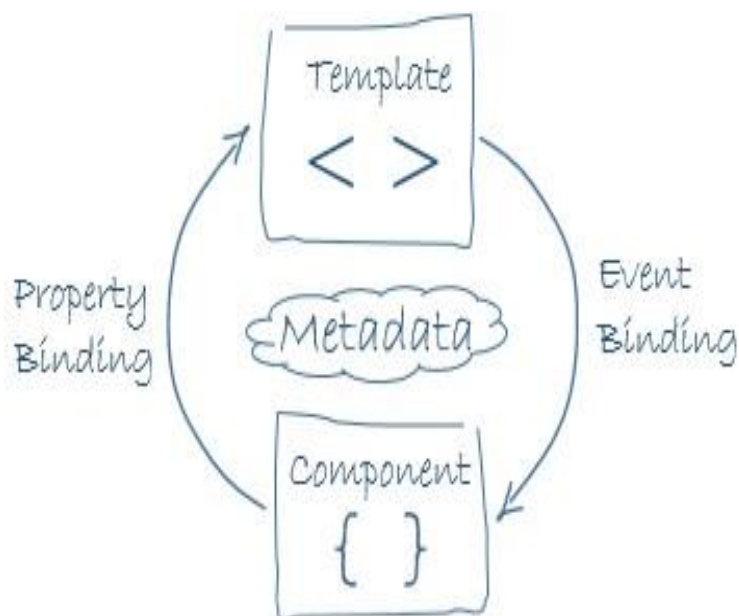


Рис. 3.19. Інкапсуляція в Angular

У Angular стилі компонента можуть бути інкапсульовані в хост-елемент компонента, щоб вони не впливали в інші компоненти. Декоратор компонента надає опцію інкапсуляції, яку можна використовувати для керування тим, як інкапсуляція застосовується для кожного компонента.

Shadow DOM схожий на паралельне дерево DOM, розміщене всередині компонента, прихованого подалі від основного дерева DOM. Жодна частина програми не має доступу до цієї тіньової DOM, крім самого компонента (рис. 3.20).

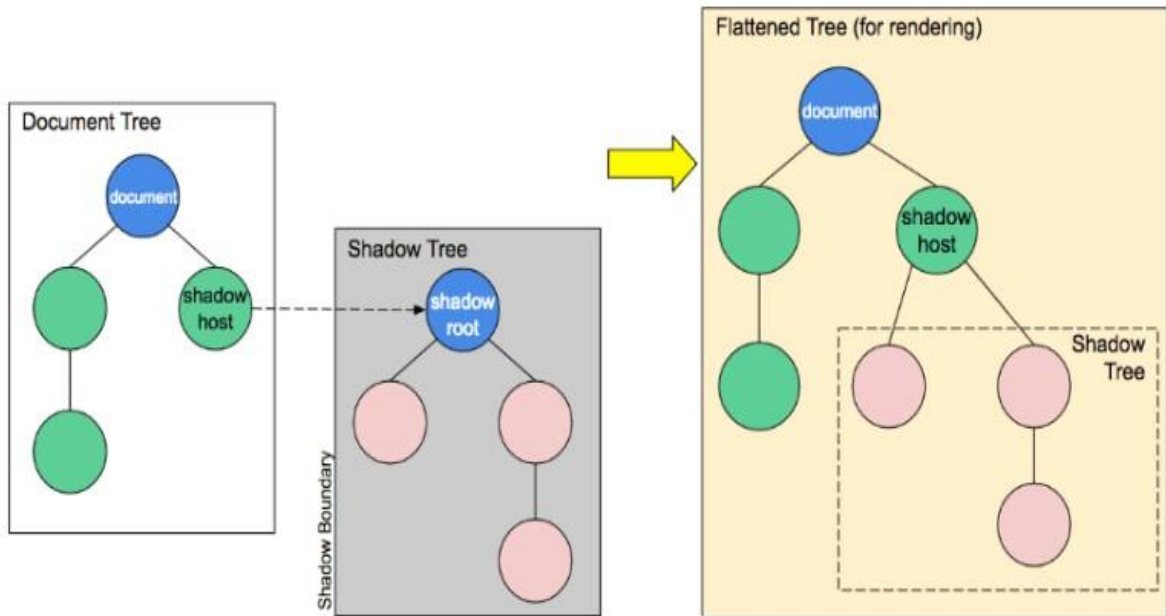


Рис. 3.20. Shadow DOM в Angular

Далі нам потрібно створити сервіси для багаторазового використання логіки (рис. 3.21). Це допоможе нам підтримувати чистоту компонентів нашого веб додатку.

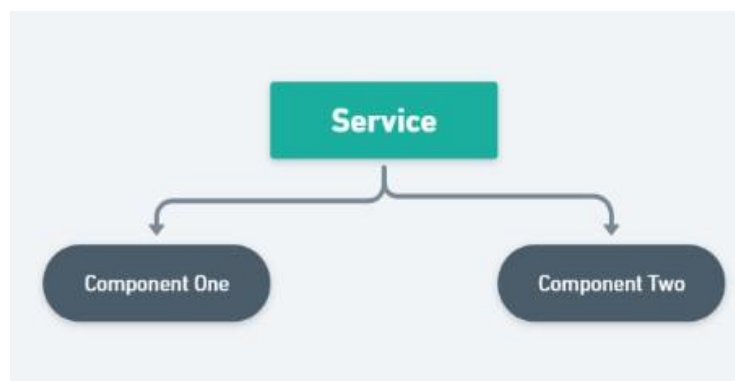


Рис. 3.21. Робота сервісів

Нам потрібно створити сервіс для отримання даних з API HTTP викликом.

Наступний крок - це робота з навігацією. Нам потрібно визначити URL адреси сторінок і переміщатися між ними в нашому додатку.

Оскільки у нас SPA додаток (рис. 3.22), маршрутизація буде здійснюватись за таким же принципом.

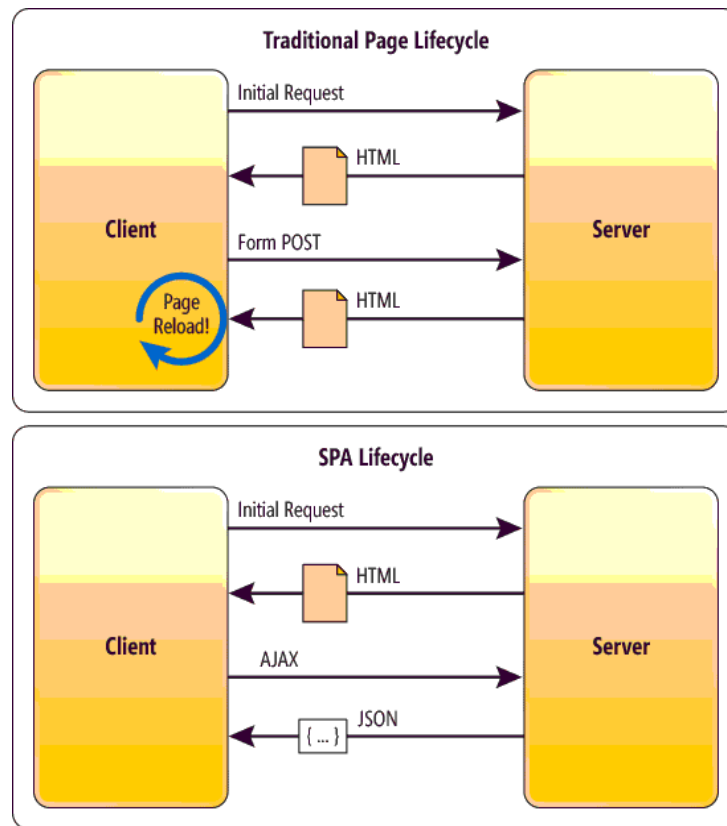


Рис. 3.22. SPA архітектура

SPA — це односторінкові програми, які завантажують одну сторінку HTML і динамічно оновлюють цю сторінку під час взаємодії користувача з програмою. SPA використовують Ajax і HTML для створення плавних і чуйних веб-програм без постійного перезавантаження сторінок.

Нам потрібно попередньо завантажити дані для сторінки, щоб не повільного завантаження даних та помилок, за допомогою обробника маршруту розв'язання.

Викорастає додатково ліниве завантаження даних різних модулів, для більш швидкої праці сайту, щоб вони не завантажувались в пам'ять, поки не знадобляться.

Треба зробити форму для збору даних з валідацією для створення конференції. Власна валідація використовує нашу бізнес-логіку для перевірки полів (рис. 3.23).

Рис. 3.23. Валідація поля у формі

Треба розширити компоненти. Це можна зробити за допомогою директив. Робимо модальний компонент, ми дізнаємось про маршрутизацію то того як це дійде до компонета, потрібно отримати дескриптор базового DOM вузла. Створюємо директиву спільно з нашим модальним компонентом.

Продовжуємо роботу з компонентою. Створюємо компоненту для голосування. Ця компонента матиме багато цікавої функціональності.

В компоненті конференції нам потрібно зробити фільтр сортування. Сортування можна назвати впорядкуванням файлів у певному порядку. Упорядкування може бути засноване на значенні кожного наявного файлу. Цей конкретний порядок може бути як зростаючим, так і спадним. Алгоритми сортування – це інструкції, які даються комп'ютеру для розташування елементів у певному порядку.

Внутрішні алгоритми сортування: це алгоритми сортування, що застосовуються до невеликої кількості даних. Використовується тільки основна пам'ять. Прикладами можуть бути бульбашкове сортування, сортування вставкою та швидке сортування.

Зовнішні алгоритми сортування: це алгоритми сортування, які можна застосувати до величезних обсягів даних. У результаті використовуються зовнішні пристрої зберігання даних, такі як жорсткі диски та флеш-диски.

Прикладом може бути сортування злиттям.

Оскільки ми плануємо розширювати веб додаток, було обрано сортування злиттям (рис. 3.24).

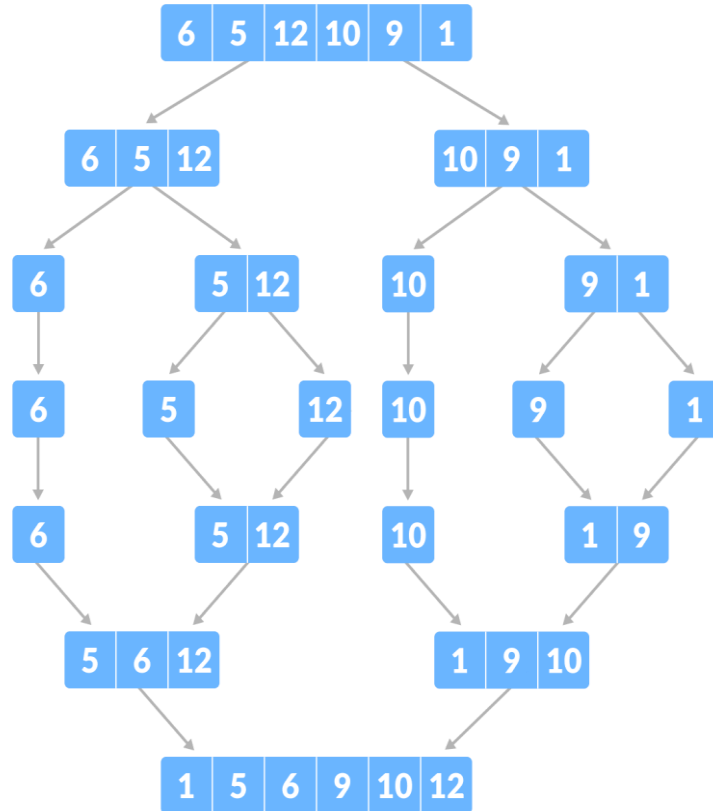


Рис. 3.24. Алгоритм сортування злиттям

Починаємо роботу з сервером.

Будемо взаємодіяти з сервером за допомогою HTTP запитів. Використовуючи RxJS, службою подій та службою виборців, змінимо функціональність, щоб зберігати дані на сервері.

Зв'язок HTTP є ключовим для веб додатку, тому цей модуль критично важливий. Треба розуміти як працюють запити (рис. 3.25).

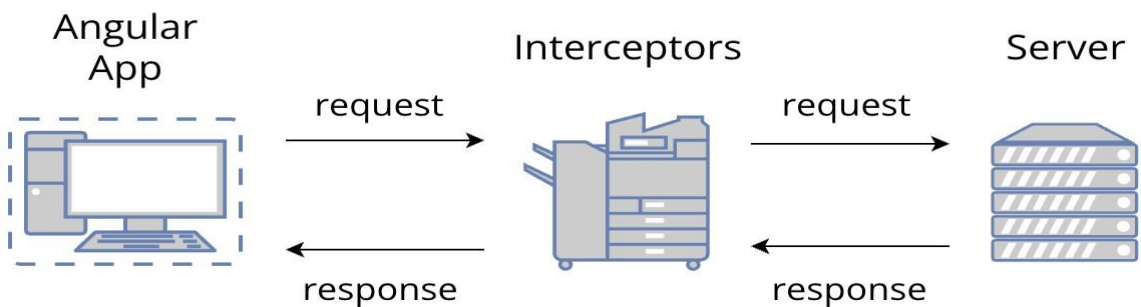


Рис. 3.25. Взаємодія HTTP з сервером

Клієнт надсилає запит на сервер, а той відповідає. Це асинхронний шаблон спілкування. Час між запитом та відповіддю може становити кілька сотень мілісекунд, тому наш клієнт продовжує обробляти, поки це відбувається.

Якщо ми не використовуємо сокети, кожен запит HTTP може представляти лише одне значення, тому можливість мати справу з потоком даних не приносить нам ніякої користі.

Він може містити кілька фрагментів даних всередині цього єдиного значення, але врешті-решт, це одна відповідь, яка повертається. Тож можливість мати справу з потоком даних не приносить нам ніякої користі.

Головним функціонал зроблено, можна переходити до тестування.

Існує досить багато варіантів при тестуванні коду. Ми почнемо із вступу до модульного тестування та деяких концепцій. Нашим основним інструментом буде Karma. Також пишемо ізольовані тести для компонентів. Ізольовані тести просто перевіряють клас компоненти, а не його інтеграцію з шаблоном.

Одиничний тест - це тест однієї одиниці коду.

Структура модульного тестування Karma дозволяє дуже і дуже чітко читати тести як речення. Крім того, чим краще ваше опис і його оператори написані, тим легше прочитати будь-які помилки, які можуть виникнути під час модульного тесту.

Щоб запустити наші тести за допомогою Angular CLI, ми використовуємо команду `ng test` у своєму терміналі. В результаті Karma відкриє браузер за замовчуванням і запустить всі тести, написані за допомогою Jasmine, і відобразить результати цих тестів.

ВИСНОВКИ

У результаті виконання даної дипломної роботи було розроблено веб додаток, за допомогою якого можна відслідковувати навчальні конференції. У роботі описується шлях створення веб додатку від ідеї, її аналізу, до втілення у життя.

1. Під час роботи над дипломним проектом, було досліджено види конференцій, знайдено найкращі шляхи відслідковування навчальні конференції.
2. Визначено наукову новизну дослідження. Актуальність теми дослідження визначає необхідність розробки веб додатку, через його розповсюдження, найкращу відповідність вимогам додатку й найбільшим потенціалом успіху в майбутньому. Для покращення розуміння можливостей та функцій, які реалізуватиме додаток, був проведений аналіз аналогів.
3. Було проведено аналіз інструментів та програмних засобів реалізації, які найбільш ефективні при створенні графічного інтерфейсу та усього функціоналу веб додатку. У ході аналізу було обрано використання мови програмування - JavaScript, Figma для створення прототипу, WebStorm як середовища розробки, Angular для односторінкової розробки веб додатку, TypeScript як надбудову для уникнення помилок при компілюванні, Karma як службу для модульного тестування, та GIT для контролю версій.
4. Визначено концепції вимог та головних елементів інтерфейсу веб додатку.
5. Розроблений детальний прототип, який дозволяє переглянути весь функціонал та графічний інтерфейс. Було створено UML діаграму структури веб додатку.
6. Розроблено графічний інтерфейс за допомогою Figma, підключення проекту до GIT, зв'язок HTTP.

7. Написання логіки додатку за допомогою мови програмування JavaScript у середовищі розробки WebStorm з використанням Angular фреймворку.
8. Розроблений додаток має сприяти самоконтролю у навчанні та мотивації користувача, покращувати навички комунікації у колективі.
9. Створений веб додаток має значні перспективи подальших досліджень. За допомогою Angular можна легко розширювати додаток, що значно збільшить кількість аудиторії. Можна доповнити функціонал, додавши можливість писати коментарі, та створити месенджер для кращої комунікації між користувачами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Dalton, N, 2001, “Fractional configurational analysis and a solution to the Manhattan problem”, in Proceedings of the 3rd International Symposium on Space Syntax, Georgia Institute of Technology, Atlanta, Georgia, pp 26.1–26.13.
2. J. Palsberg and Z. Su (Eds.): SAS 2009, LNCS 5673, pp. 238–255, 2009. c Springer-Verlag Berlin Heidelberg 2009.
3. Fielding, Roy T.; Gettys, James; Mogul, Jeffrey C.; Nielsen, Henrik Frystyk; Masinter, Larry; Leach, Paul J.; Berners-Lee, Tim (June 1999).
4. ["Angular, version 2: proprioception-reinforcement"](#) September 14, 2016. Adam Nathan Universal Windows Apps with XAML and C# Unleashed /- Sams Publishing – c. 104 – 130.
5. Demetrescu, Camil; Finocchi, Irene; Ribichini, Andrea (22 October 2011), "Reactive Imperative Programming with Dataflow Constraints", [Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications](#), Oopsla '11, pp. 407–26.
6. "Karma" in: John Bowker (1997), The Concise Oxford Dictionary of World Religions, Oxford University Press, pp 351–352.
7. Erik Zurcher (1980), Buddhist influence on early Taoism, T'oung Pao, Vol. 66, pp 84–147.
8. Yuvraj Krishan (1985), The doctrine of Karma and Śraddhas, Annals of the Bhandarkar Oriental Research Institute, Vol. 66, No. 1/4, pages 97–115.
9. See Kumarila's Slokavarttika; for English translation of parts and discussions: P. Bilimoria (1990), "Hindu doubts about God – Towards a Mimamsa Deconstruction", International Philosophical Quarterly, 30(4), pp. 481–499.

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА FRONT END ЧАСТИНИ ПЛАТФОРМИ ДЛЯ
КОНФЕРЕНЦІЙ НА ANGULAR ФРЕЙМВОРКУ

Виконала студентка 4 курсу

Групи ПД-44

Вероївська Анастасія Олексіївна

Керівник роботи

Корецька В.О.

Київ - 2022

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

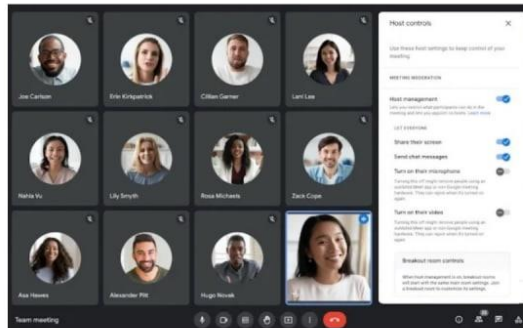
- **Мета дослідження** - покращення процесу проведення веб конференцій за допомогою розробленого веб додатку для конференцій на Angular фреймворку.
- **Об'єкт дослідження** - процес проведення конференцій.
- **Предмет дослідження** - сучасні технології в процесі проведення конференцій.

АКТУАЛЬНІСТЬ РОБОТИ

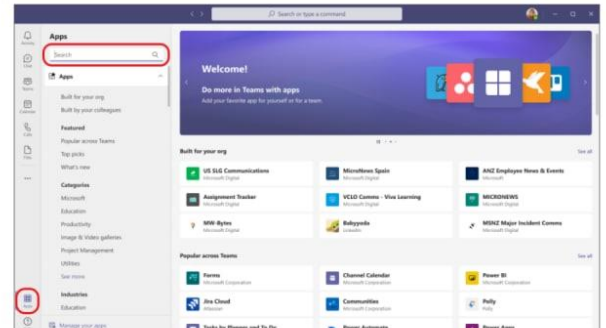
- Актуальність роботи полягає в тому, що розроблений веб додаток допомагає ефективніше навчатися та відслідковувати майбутні онлайн конференції.
- Веб-конференції – це простий спосіб раціонального використання робочого часу. Тому надзвичайно актуальним є розробка веб додатку для проведення конференцій що сприятиме самоконтролю та мотивації користувача в навчальному процесі, покращить навички комунікації у колективі.

3

НАЙБІЛЬШ ПОДІБНІ ДО АНАЛОГІВ



Google Meet



Microsoft Teams

4

ПОРІВНЯННЯ З АНАЛОГАМИ

Назва	Переваги	Недоліки
Google Meet	• Наявність функції чату	• Відсутність запису зустрічі
	• Студенти можуть створити сесію	• Відсутність прямої трансляції
		• Можливість обміну файлами
Microsoft Teams	• Можливість запису зустрічі	• Відсутність збереження запису у хмарі
	• Можливість обміну файлами	• Велика кількість користувачів одночасно

5

ТЕХНІЧНЕ ЗАВДАННЯ

- Дослідити актуальність прецесу відслідковування веб конференцій.
- Розробити архітектуру веб додатку.
- Розробити прототип.
- Реалізувати можливість відображення веб конференцій.
- Реалізувати можливість оформляти нову коференцію для користувача.
- Реалізувати можливість для користувача продивлятися сесії у данній конференції.
- Реалізувати можливість фільтрації сесій по рейтингу та рівню складності.
- Реалізувати можливість змінювати данні користувача у профілі.

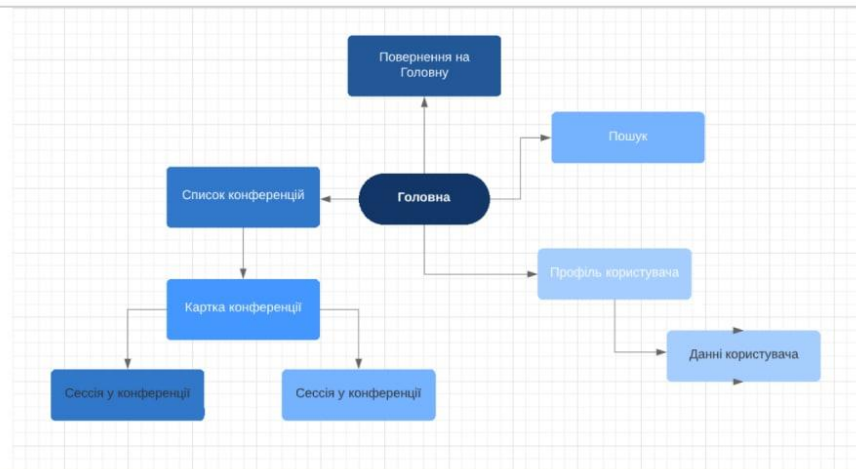
6

ПРОГРАМНІ ТА ТЕХНІЧНІ ЗАСОБИ РЕАЛІЗАЦІЇ



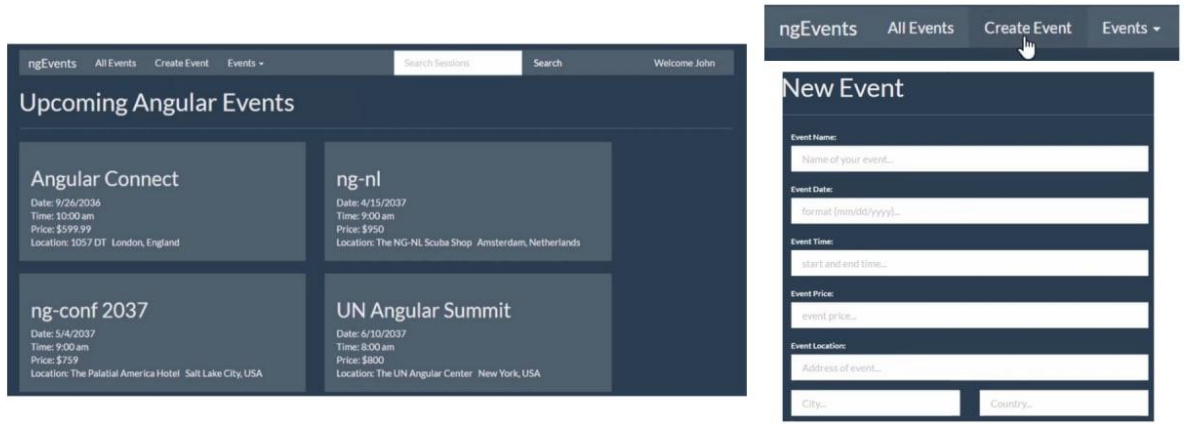
7

АРХІТЕКТУРА ВЕБ-ДОДАТКУ

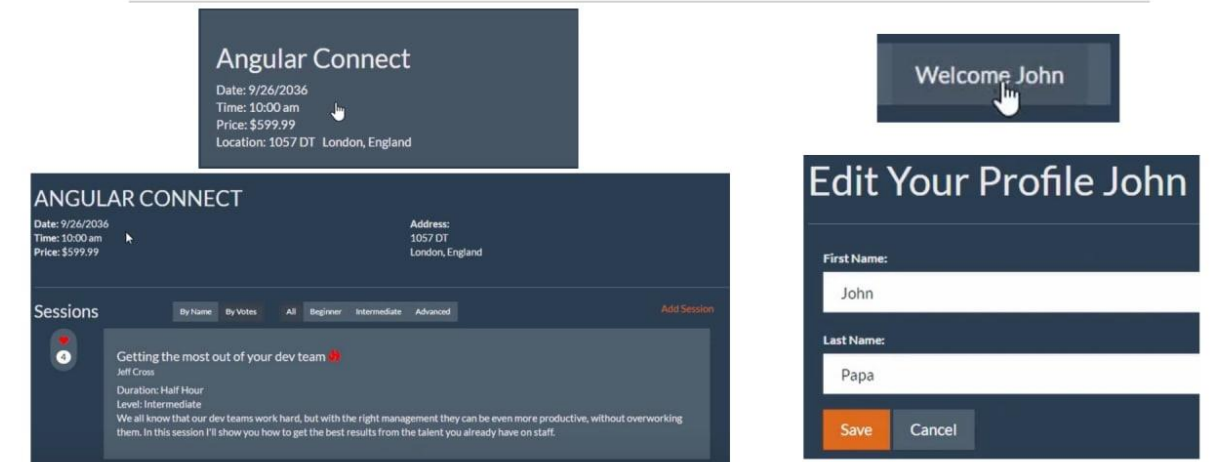


8

ЗОВНІШНІЙ ВИГЛЯД ВЕБ-ДОДАТКУ



ЗОВНІШНІЙ ВИГЛЯД ВЕБ-ДОДАТКУ



ВИСНОВКИ

Досліджено:

- Процес проведення веб конференцій.
- Сучасні технології проведення веб конференцій.

Розроблено:

- Архітектура веб додатку для відслідковування веб конференцій.

Реалізовано:

- Можливість відображення веб конференцій.
- Можливість оформляти нову конференцію для користувача.
- Можливість для користувача продивлятися сесії у данній конференції.
- Можливість фільтрації сесій по рейтингу та рівню складності.
- Можливість змінювати данні користувача у профілі.

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Вероївська А.О., АКТУАЛЬНІСТЬ СТВОРЕННЯ ВЕБ ДОДАТКУ ДЛЯ ВІДСТЕЖЕННЯ ПРОВЕДЕННЯ КОНФЕРЕНЦІЙ / НАУКОВО-ТЕХНІЧНА КОНФЕРЕНЦІЯ "ЗАСТОСУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ІНФОКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЯХ" Збірник тез 20.04.2022, ДУТ, м. Київ – К.: ДУТ, 2022.

ДЯКУЮ ЗА УВАГУ!

