

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ФУНДАМЕНТАЛЬНИХ
НАУК
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ

Курсова робота
з курсу «Програмування Web-додатків»

Варіант № 1

Виконав:
Студент групи ПМ-33
Венгринюк Олег

Перевірив:
Топилко П.І.

Львів 2019

ЗМІСТ

1.	Вступ.....	3
2.	Розділення шаблону веб-сторінки на логічні компоненти.....	5
3.	Створення статичних елементів.....	6
4.	Створення динамічних (інтерактивних елементів).....	8
5.	Висновки.....	9
6.	Список використаних джерел.....	10
7.	Додаток А.....	11

ВСТУП

В період стрімкого розвитку цифрових технологій, розроблення веб додатків є дуже популярною галузю інформаційних технологій. Є дуже багато написаного відкритого програмного забезпечення та велика кількість спеціалістів, що створює потужну базу для створення web продукту.

Розроблення застосунків за допомогою web-технологій робить застосунок інверсальним та дешевим у розробці і підтримці.

На даний момент є багато різновидів web-сайтів. До прикладу, односторінкові веб-додатки (Single Page Application) є прикладом інтерактивних сторінок, який дуже схожий на звичайний додаток, при цьому цей веб-додаток зберігає переваги веб-сайтів. Основна ідея односторінкових додатків – це те, що сторінка завантажується лише один раз, а все інше відбувається за рахунок запитів.

Мета цієї роботи є вивчення теоретичної бази односторінкового додатку та розробка клієнтської частини односторінкового додатку, використовуючи фреймворк React.

Звичайні web-сторінки прянують за допомогою постійних запитів на сервер. Так вдається відокремити серверний функціонал від графічного інтерфейсу користувача. Це відбувається наступним чином: користувач посилає через web-браузер get-запит і отримує http-відповідь (response англ.) і опрацьовує та відображає відповідь належним чином.

Основні переваги односторінкових додатків:

1. Простота інтерфейсу користувача.
2. Зменшення ролі сервера.
3. Спрощення логіки сервера.
4. Можливість використання багатьох серверних додатків.

В той час є певні недоліки:

1. Збільшення навантаження на клієнтський застосунок.
2. Складність розробки.

Односторінкові сайти (landing page) складається з функціональних або стильових компонент. Компонента – об'єкт, який має внутрішній стан та відображення, компонента може включати інші компоненти або бути включеною в них.

При розробці клієнтської частини веб-додатку часто використовують шаблони MVC (Model-View-Controller) та MVVM (Model-View-ViewModel). Концепція MVC полягає у розбитті коду на наступні логічні частини: модель (model) – дані та логіку роботи з ними; представлення (view) – модель даних, її представлення та контроллер (controller) – логічна частина, яка приймає введення та оновлення моделі. MVVM – новіша версія шаблону MVC яка є

адаптована під сучасні методології розробки. Основні логічні компоненти MVVM – це модель (Model), так само як в MVC – представляє дані та логіку роботи з ними; представлення (View) – логічна частина, яка відображає модель та модель представлення (ViewModel) – абстрактне відображення представлення.

Для виконання завдання цієї роботи було використано фреймворк React. React – це популярний фреймворк для розробки односторінкових додатків, створений компанією Facebook. Основні можливості та переваги Angular

1. Простота розробки
2. React - це бібліотека для створення користувацьких інтерфейсів
3. Підтримка сучасних шаблонів проектування.
4. Підтримка компонентної архітектури додатків.
5. Легке впровадження залежностей.
6. Відкритий код, який розповсюджується за MIT ліцензією.

У ході роботи був розроблений односторінковий веб-додаток на прикладі якого були розглянуті особливості розробки з використанням React.

React побудований на принципі компоненти — частини користуцького інтерфейсу. Компонети контролюють DOM – елементи та описують логіку відображення та поведінки.

На даний момент React є бібліотекою з широким функціоналом, що не потребує великих обчислювальних витрат в порівнянні з фреймворками.

Розділення шаблону веб-сторінки на логічні компоненти

Наданий макет було розділено на 9 логічних незалежних компонент. Серед цих компонент: зображення з логотипом та аватар користувача; навігація сайту; меню налаштування інтерфейсу користувача; календар; налаштування безпеки; інформаційна стрічка з наявністю суманих продажів; кількістю замовлень; кількість продукцій; кількістю нових замовлень.

На основній сторінці компоненти були поміщені у сітку, створену з використанням CSS Flexbox.

За функціоналом та особливістю його роботи елементи можна поділити групи: статичні елементи та динамічні елементи. Статичні елементи – це елементи, які не змінюють свого вигляду протягом перегляду сторінки. Динамічні елементи – це елементи з якими можна взаємодіяти під час відвідування сайту, і які в залежності від дій користувача можуть міняти свою поведінку.

Створення статичних елементів

До статичних елементів на реалізованій сторінці відносимо: зображення з логотипом та інформаційний рядок; навігація по розділах сайту; меню для редагування користувацької сторінки та відповідні показники трафіку і голосувань користувачами; вікно з описом власника сайту.

Статичний елемент № 1 – зображення з логотипом. Для простоти було використано доступний html контейнер `<div>` та створенно графічний об'єкт за допомогою елемента `img`. (рис 1)

Статичний елемент № 2 – навігація сайту. Компонент, що описаний у файлі **MenuItem.jsx**. В ньому вказані основні css властивості для коректного відображення на сторінці користувача. Спочатку ескартинка пошуку та можливість сховати підземню для редагування інтерфейсу та аватру власника сайту. Тобто статичні елементи 3 та відповідно 4. (рис 2)

Статичний елемент № 3 – аватар власника сайту з його прізвищем та ім'ям. Для спрощення було використано звичайни контейнер з html та елемент для представлення графічного об'єкту, а саме тег `img`. В цього елемента на картинці видно коли користувач є доступний онлайн чи відсутній. Також біля аватару власника сайту присутнє його ім'я та місце проживання. Що зручно для відвідувачів сайту. (рис 3)

Статичний елемент № 4 – підменю для редагування інтерфейсу. Підменю зміщає в собі вкладення декількох однакових зі сторони верстки об'єктів. В підменю є можливість переходу до дошки (Dashboard) а також до повідомлень (Messages). Також передбачено згідно з макетом підменю для редагування елементів графічного інтерфейсу. І звісно ж календар. Також доступні з підменю перехід до кабінету користувача, налаштувань безпеки та інформації про споживачів. Меню відповіде на натискання лівою клавішою мишки користувачем по відповідному елементу меню. (рис 4)

Статичний елемент № 5 – інформаційний рядок. Компонента, що описана у файлі **Card.jsx**. І є компонентою **SmallCard**. Вона предсталає собою картинку з текстовою інформацією про кількість продажів, нотатків, замовлень, продукції. (рис 5)

Статичний елемент № 6 - діаграма кількості відвідувачів та зареєстрованих користувачів. Компонента описана у файлі **WebsiteTraffic.jsx** під назвою **WebsiteTraffic**. Дана компонента є композитною, оскільки включає html елементи, що були використані для представлення текстової інформації та компоненту, описану в файлі **WebTrafficPie.jsx** під назвою **WebTrafficPie**. Саме компонента **WebTrafficPie** відповідає за представлення діаграми. (рис 6)

Статичний елемент № 7 — інформація про власника сайту з вказаною кількістю його клієнтів, продукції та підписників. Дана компонента описана в файлі **User.jsx** під назвою **UserCard**. Вона включає в себе графічну складову та

текстово-інформаційну. До графічної входять: фотографія власника сайту, фонові фотографії. До текстово-інформаційної: цитата та кількість споживачів, продукції та підписників. (рис 7)

Статичний елемент № 8 — інформаційний компонент задоволеності користувачів. Даний компонент описаний у файлі **CustomerRatingProgressbar.jsx** під назвою **CustomerRatingProgressbar**. В ній міститься інформація з графічним представленням у вигляді прогрес-бару про задоволеність користувачами продукцією. (рис 8)

Створення динамічних (інтерактивних елементів)

Інтерактивні елементи – це елементи, що взаємодіють з користувачем і залежно від його поведінки змінюють своє предсталення.

Інтерактивний елемент №1 - графік відвідування сайту. Елемент що описаний у файлі `ProductVisitiors.jsx` компонент під назвою **ProductVisitor**. (рис 9)

Інтерактивний елемент №2 -графік загальних витрат та придутку. Ця логічна компонента описана у файлі **Card.jsx** під назвою **SmallSquareCard** (рис 10)

Інтерактивний елемент №3 - тижневих продажів та витрат. Ця логічна компонента описана у файлі **Card.jsx** під назвою **SmallSquareCard** (рис 11)

ВИСНОВКИ

В ході виконання курсової роботи, в першому розділі було на практиці опановано створення односторінкових веб-сторінок. Було також освоєно основні складові легкого фреймворку React.

В основній частині було описано хід роботи, мотивацію та причини вибору шаблону, принцип поділу сторінки на компоненти, опис реалізації кожної компоненти та функціонал та бібліотеки використані для реалізації функціоналу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Official React Tutorial: Intro to React
<https://reactjs.org/tutorial/tutorial.html>
2. Start Learning React
<https://egghead.io/courses/start-learning-react>
3. The Beginner's Guide to React
<https://egghead.io/courses/the-beginner-s-guide-to-react>
4. The Road to learn React
<https://leanpub.com/the-road-to-learn-react>
5. React Tutorial: A Comprehensive Guide to learning React.js in 2018
<https://tylermcginnis.com/reactjs-tutorial-a-comprehensive-guide-to-building-apps-with-react/>
6. w3school html guide
<https://www.w3schools.com/html/>
7. w3school css guide
<https://www.w3schools.com/css/>
8. w3school css FlexBox
9. https://www.w3schools.com/css/css3_flexbox.asp

ДОДАТОК А



Рис 3

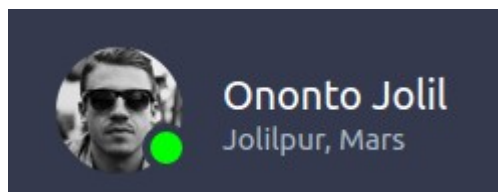


Рис 1

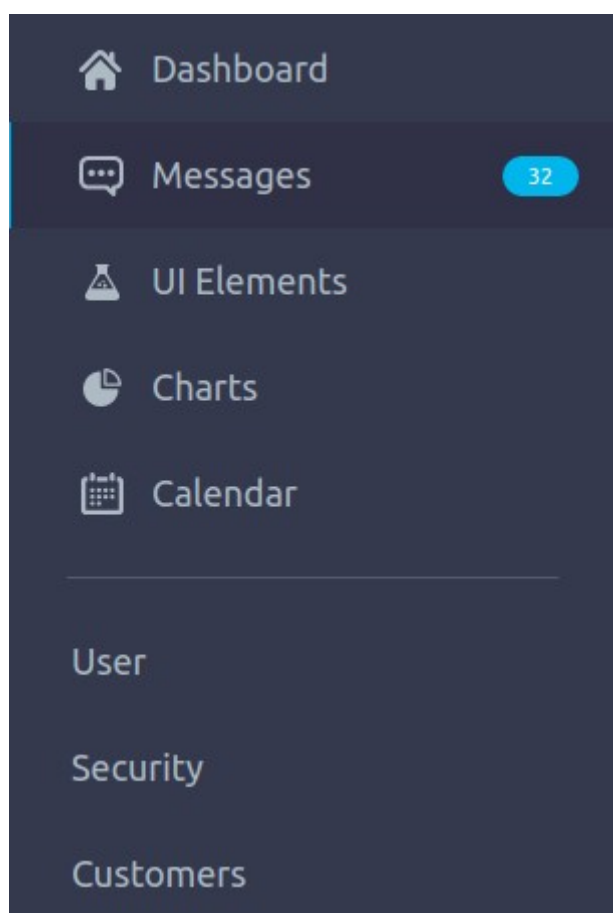


Рис 4

Рис 2

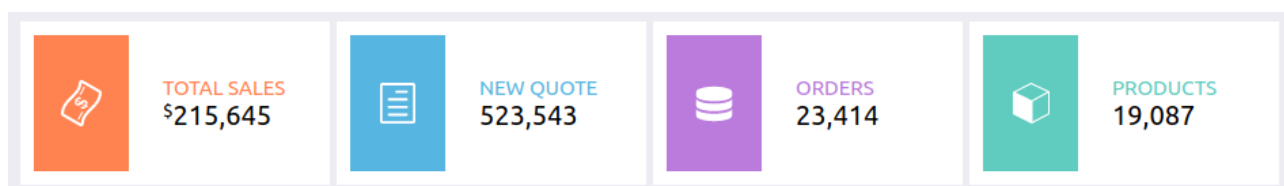


Рис 5

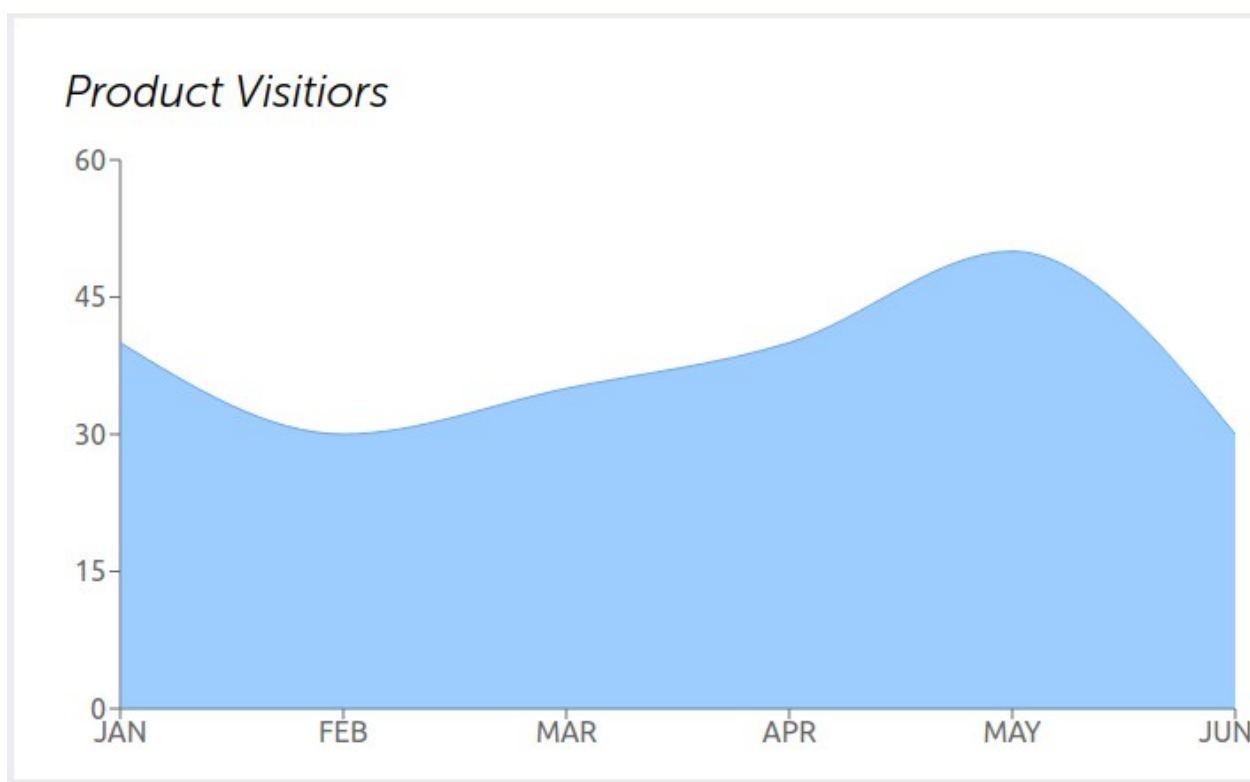


Рис 11

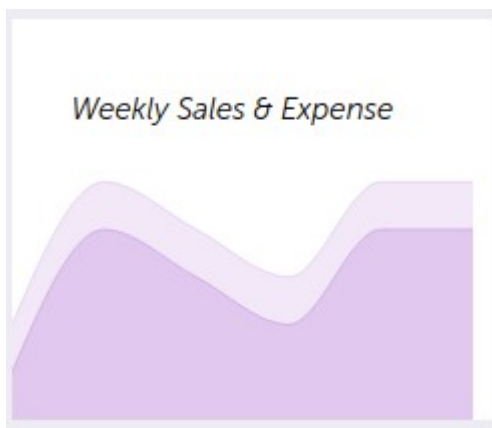


Рис 9

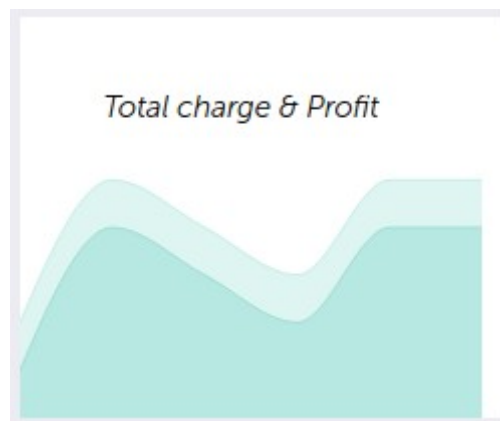


Рис 10

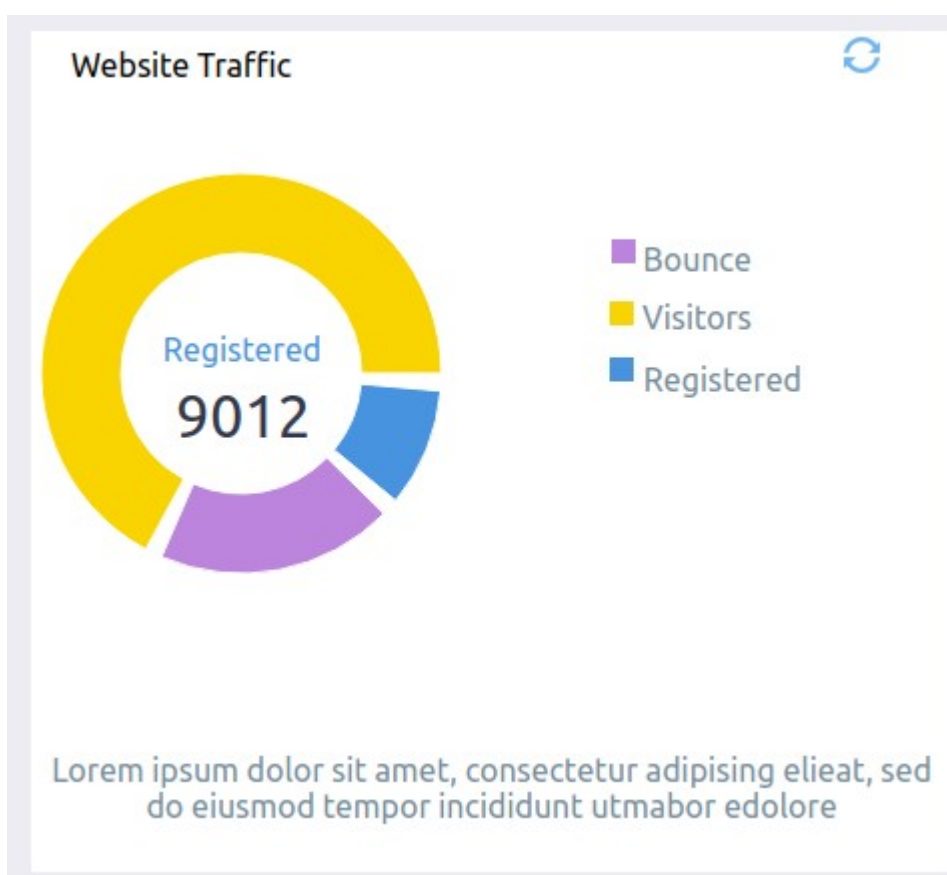


Рис 6



Рис 7

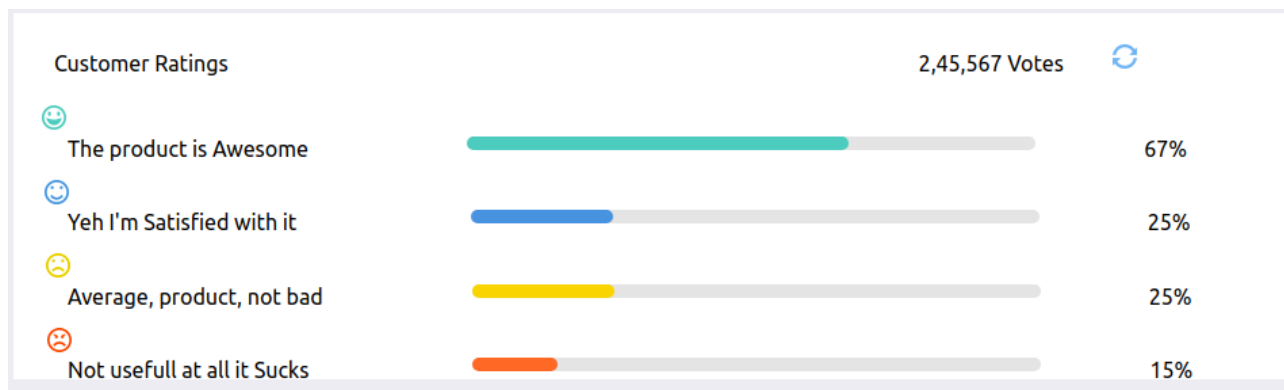


Рис 8