МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
(назва факультету)

КАФЕДРА КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
(повна назва кафедри)

ЗВІТ

З ВИРОБНИЧОЇ ПРАКТИКИ

Керівник від бази практики:
Директор з персоналу Делпорте Р.С.
(посада, підрозділ, прізвище та ініціали)

(підпис) МП

Студента (ки) 3 року навчання
Групи 6.04.125.010.21.2
першого (бакалаврського) рівня вищої освіти
спеціальності 125 "Кібербезпека"
ОПП "Кібербезпека"
Бойко Вадим Віталійович
(прізвище та ініціали)

Керівник від ЗВО:
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Кількість балів з урахуванням захисту _____
Національна шкала _____ Оцінка: ECTS _____

Члени комісії:
(підпис) (прізвище та ініціали)
(підпис) (прізвище та ініціали)
(підпис) (прізвище та ініціали)

м. Харків, 2023 р.

Завдання на виконання практики

ЗАВДАННЯ НА ВИРОБНИЧУ ПРАКТИКУ

(вид практики)

1. Назва завдання: проходження практики на ТОВ «Українські Інформаційні технології»
2. Вхідні дані до завдання: проходження практики на ТОВ «Українські Інформаційні технології»

Керівник від ЗВО _____

(підпис) (посада, П. І. Б.)

Студент _____

(підпис) (П. І. Б.)

Зміст

ВСТУП

На сьогоднішній день світова павутина під назвою інтернет має дуже величезну кількість людей, що її використовує. Коли інтернет тільки зароджувався і думки не було що в майбутньому буде настільки великий прогрес у даній сфері, що будуть з'являтися такі дивні штуки як сайти, додатки та інші новинки нашого часу.

На той час такого поняття як сайт не існувало, був просто макет і все .Професія Верстальник сайтів може дуже сильно знадобитися в сучасному світі так як вона є чи не найбільш затребуваною професією програмістів. А що таке верстка сайтів? Чим є дана професія, важко її отримати і важко все запам'ятати. Коротко кажучи Не так вже й просто.

Основним завданням практики є показати що таке верстка сайтів, як її робити, як створювати та змінювати сайти. Це корисна річ

1.Характеристика бази практики

Об'єктом проходження практики є розробка веб компонентів у компанії SoftServe.

Під час проходження практики було створено декілька веб компонентів та проаналізовано які методи стосовно безпеки впроваджені на проекті

Кожен компонент тестується QA та перевіряється на відповідність до критеріїв приймання, й у разі не відповідності компонент повертається для доопрацювання

SoftServe

SoftServe — українська IT-компанія, що працює у сфері розробки програмного забезпечення та надання консультаційних послуг.

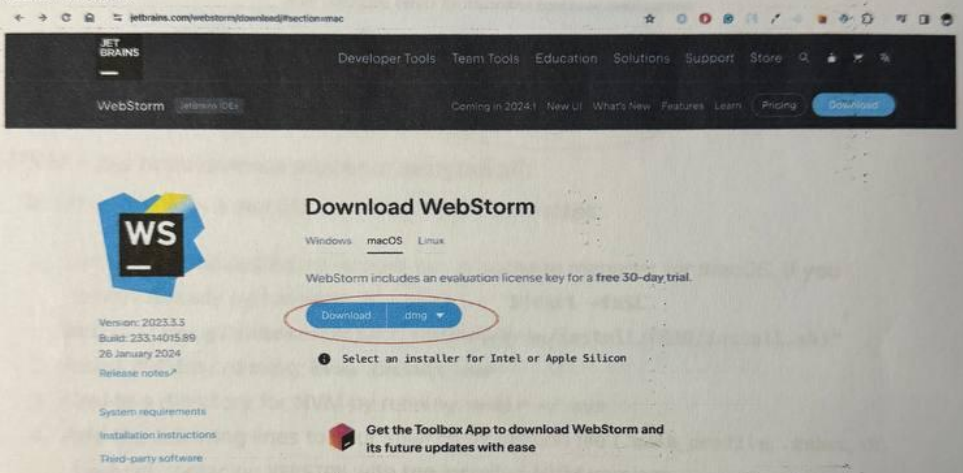
Головні офіси компанії розташовані у Львові й Остіні (штат Техас, США), понад 10000 працівників працюють в офісах компанії в Європі, США, Канаді, Латинській Америці, Сінгапурі і Дубаї. SoftServe є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі та є другою найбільшою IT-компанією України за кількістю співробітників.

2. Підготовка до роботи

Перед початком роботи потрібно встановити необхідне ПЗ. У моєму випадку це WebStorm як програма для написання коду, Git для роботи з версіонуванням коду, Docker для запуску середовища в однакових умовах на будь якому девайсі, та nvm для зручного перемикання між версіями мови node.js

2.1 Встановлення

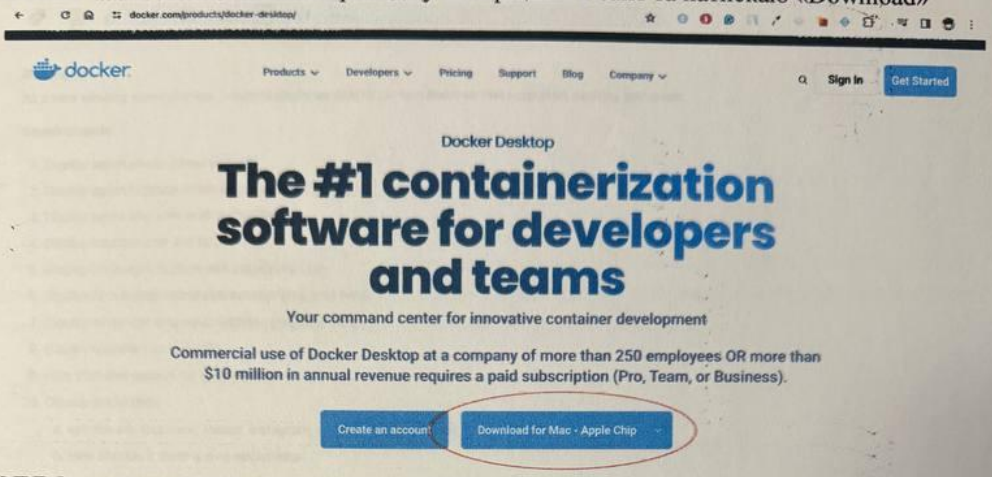
WebStorm – для встановлення потрібно перейти на сайт продукту та натиснути «Download»



Git – для встановлення потрібно виконати наступну команду в терміналі

```
$ brew install git
```

Docker – для завантаження переходжу на офіційний сайт та натискаю «Download»



NVM – для встановлення виконую наступні дії:

To set up NVM on a macOS machine, follow these steps:

1. Open Terminal and install Homebrew, a package manager for macOS, if you haven't already by running: **`/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`**
2. Install NVM by running: **`brew install nvm`**
3. Create a directory for NVM by running: **`mkdir ~/.nvm`**
4. Add the following lines to your shell configuration file (**`.bash_profile`**, **`.zshrc`**, or **`.bashrc`**), replacing **`VERSION`** with the installed NVM version:

```
export NVM_DIR="$HOME/.nvm"
[ -s "/usr/local/opt/nvm/VERSION/nvm.sh" ] && . "/usr/local/opt/nvm/VERSION/nvm.sh"
```

5. Close and reopen Terminal, or run **`source ~/.bash_profile`**, **`source ~/.zshrc`**, or **`source ~/.bashrc`** to reload the shell configuration.
6. Verify that NVM has been installed correctly by running: **`nvm --version`**

2.2 Огляд задачі

На проекті програмою для відстеження задач є JIRA, тому переходжу та дивлюсь на задачу
Однією з задач була розробка компонента з даними вимогами:

Web - Agent profile top card

Description

As a user viewing agent profiles, I need to easily be able to contact them so that I can start working with them.

Requirements

1. Display agent photo (required field)
2. Display agent first/last name (required field)
3. Display agent title with team name/dba
4. Display location icon and location name
5. Display languages spoken and associated icon
6. Display phone icon and phone number (required field)
7. Display email icon and email address (required field)
8. Display website icon and URL
9. Hide icon and section for empty fields
10. Display social links
    a. options are facebook, twitter, instagram, youtube and/or linkedin
    b. hide section if there are no social links
11. Display License # and state abbreviation
    a. could be more than one - need to show entirety
12. Display market center name
    a. could be more than one - need to show entirety
13. Connect with CTA will be handled in another story
14. Luxury pill if applicable

Також до задачі був доданий дизайн за яки м потрібно було створювати компонент

3. Процес роботи

Для початку потрібно створити нову гілку у git-у,
для цього виконаю наступну команду

git checkout -b <назва гілки>

3.1 Встановлення пакетів

Перед початком розробки компонента потрібно впевнитись, що пакети встановлені та актуальні, відповідно до проекту, для пакетного менеджеру використовую вбудований для node.js менеджер – npm, та виконую наступну команду

npm ci

що допоможе встановити або оновити пакети до необхідної версії

3.3 Запуск BE сервісів локально

Для запуску BE для локальної роботи використвоую Docker
Всі параметри для запуску сервісів знаходяться у файлі «docker-compose.yml»

Для запуску потрібно виконати наступну команду
docker-compose up

3.4 Запускаю FE частину, бо без BackEnd частини воно не працює

Для цього у проекті у файлі «package.json» є розділ «scripts» де можна побачити вже створені скрипит.

В моєму випадку для запуску я виконав команду «npm run dev»

Після виконання команди переходжу на localhost та у вкладці network бачу, що помилок не має, отже проект запущено правильно

4. HTML&HBS

На даній частині проекті впроваджена логіка server side рендерінгу, де основна ідея полягає у тому, що ми використовуємо темплейти, та передаємо дані до них, вони отримавши дані відображають їх у вигляді звичайного HTML коду

4.1 Додав всі необхідні дані до .hbs файлу

Для локального тестування потрбно або використовувати файли типу .json з усіма необхідними параметрами, або брати дані динамічно з BE сервісів

4.2 HTML5 Layout

Коли всі необхідні дані були передані до темплейту створюю верстку та димачно відображаю або ховаю елементи в залежності від даних, які були передані до темплейту

5. CSS&SCSS

Наступним кроком додаю стилі до компоненту, на проекті використовується SCSS, що дозволяє розширити функціонал звичайного CSS-у,

Sass (англ. Syntactically Awesome Stylesheets) — скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS). Спроектована Гемптоном Кетліном та розроблена Наталі Вейзенбаум. Sass призначений для підвищення рівня абстракції коду та спрощення файлів CSS.

Мова Sass має два синтаксиси:

- sass (оригінальний) — відрізняється відсутністю фігурних дужок, в ньому вкладені елементи реалізовані за допомогою відступів, а правила відокремлюються переведенням рядка;
- scss (новий) — використовує фігурні дужки (подібно до CSS).

5.1 Адаптивна верстка

Коли компонент готовий для одного типу екрану починаю додавати адаптивну верстку,

На проекті використовується підхід mobile first, тому спочатку була зроблена мобільна версія компоненту, а потім завдякі медіа запитам було додано адаптивну версію компонету,

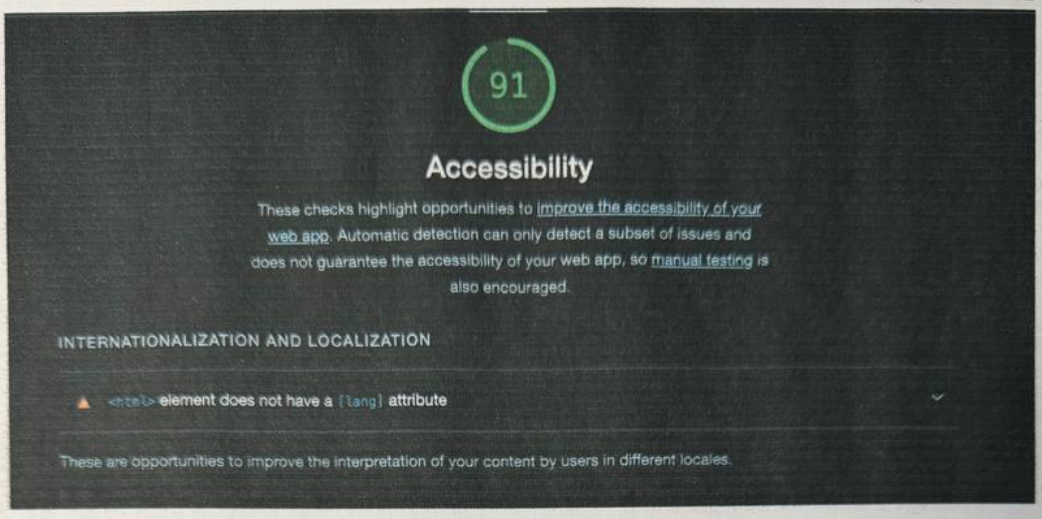
Медіазапит — це функція в CSS3, яка адаптує макет сторінки до різних розмірів екрана та типів медіа.

Синтаксис:

```
@media media type and (condition: breakpoint) {
  // CSS rules
}
```

5.2 Перевірка компонента на доступність для людей з обмеженими можливостями

На проекті використовується утіліта «Lighthouse» завдяки якій можна перевірити на скільки компонент доступний для людей з обмеженими можливостями, на скільки сайт має SEO оптимізацію та швидкість завантаження, про те нам цікава лише вкладка стосовно обмежених можливостей та на скільки в нас доступний компонет



Можна побачити, що зауважень до розроблємого компоненту не має, отже можна віддавати код на тестування, про те спочатку код має отримати дозвіл від інших розробників для цього потрібно надіслати код до GitHub-у

6. Git&GitHub

Git — розподілена система керування версіями файлів та спільної роботи. Проєкт створив Лінус Торвальдс для керування розробкою ядра Linux, а сьогодні підтримується Джуніо Хамано (англ. Junio C. Hamano). Git є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні та злитті гілок.

GitHub — один з найбільших вебсервісів для спільної розробки програмного забезпечення. Існують платні та безплатні тарифні плани користування. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc

6.1 Створення гілки

Для створення гілки скористаюсь командою «git checkout -b <branch_name>»,

Проте на проекті є правила створення гілки, тому будемо слідувати ним,

а саме – гілка має називатись <feature|bugfix/ticket_number/message>

Створивши гілку потрібно додати фали

6.2 Створення коміту

Коміт у git — це об'єкт, що містить посилання на:

- знімок стану (snapshot), що записаний раніше в індекс
- метадані автора
- коментарі
- нуль чи більше вказівників на інші коміти, що є прямими батьками даного коміту

Для додавання коду до git-у виконаю команду git add

6.3 Надсилання коду до GitHub-у

Для надсилання коду потрібно виконати наступну команту

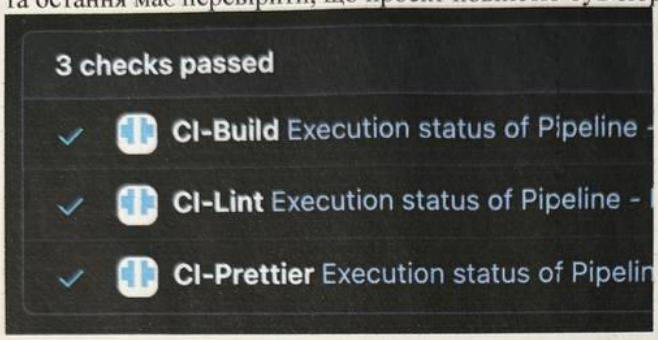
git push

після чого код буде відправлений до системи GitHub,

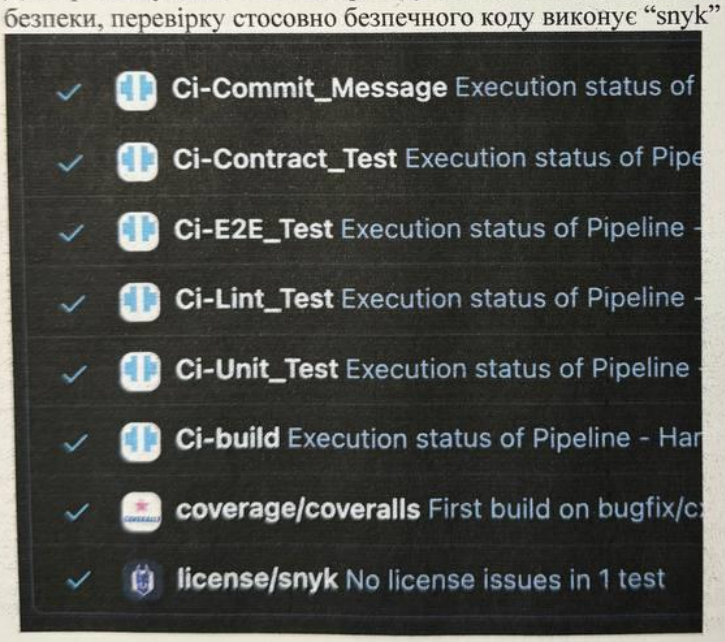
Де ми зможемо отримати коментарі інших розробників та отримати дозвіл на злиття до основної гілки

6.4 Merge checks

Кожен запит на злиття коду має пройти через перевірку до злиття коду, так на даній частині проекту 3 перевірки – одна для стилю написання коду, інша для форматування та остання має перевірити, що проект повністю був зібраний



Для прикладу інша частина проекту має більше перевірок, та одна з них стосується безпеки, перевірку стосовно безпечного коду виконує “snyk”



6.5 Тестування

Після отримання дозвілу на зливання коду до основної гілки статус задачі змінюється на “Ready for QA” та очікує на тестування,

якщо все добре, тоді статус змінюється на «Ready for Release», якщо є щось, що потрібно допрацювати – задача повертається назад до розробника

7.Результати моєї роботи на практиці

Ця практика є дуже потрібною для початку та навіть для більш знаючих людей може бути корисна для закріплення на практиці чи вивчення нових технологій розробки та правил чистого коду. Також для мене був дуже корисний Git допоміг не просто прочитати якийсь матеріал , а ще й самому створити частини сайту і побачити результат. Сам я зробив для себе замітки про принципах чистого коду щоб в подальшому я міх програмувати більш якісно.

8.Висновок

В цілому можу сказати що ця практика була корисна для мене я багато чого вивчив та запам'ятав. Я зрозумів як побудована логіка, структура сайтів, а також навчився її створювати. Вивчив багато допоміжних технологій та зрозумів як працюють між собою ріні частини застосунку

Список використаних джерел

1. **https://uk.wikipedia.org/wiki/SoftServe**
2. **https://www.atlassian.com/git/tutorials/install-git**
3. **https://www.docker.com/products/docker-desktop/**
4. **https://codedamn.com/news/nodejs/nvm-setup-guide**
5. **https://uk.wikipedia.org/wiki/Sass**
6. **https://uk.wikipedia.org/wiki/GitHub**
7. **https://uk.wikipedia.org/wiki/Git**