**Звіт 1(фронт)**

Так як і з беком, я поділив фронт-проект на частини:

* логін/реєстрація нових користувачів
* Основна сторінка (Тут теж буде розділення на частини)
* Підключення React Leaflet
* Зв’язування з беком (Цей етап умовний, бо буду робити «зв’язування» по мірі необхідності та готовності компонентів)

Я вже створив сторінку логіну/реєстрації та зв’язав її з бек-енд частиною.

Оскільки я використовую фреймворк **React**, то для відображення сторінки я створив компонент **SignInUp** у якому визначив метод **render()**, який повертає jsx елемент, що конвертується у html. Також я написав два методи, **doLogin** та **doSignUp**, які викликаються при підтвердженні відповідних форм логіну або реєстрації. У самому компоненті я визначив елемент state:

state = {

login: true,

signUpForm: {

firstName: "",

lastName: "",

email: "",

phone: "",

password: "",

passwordRep: ""

},

signInForm: {

email: "",

password: ""

},

token: ""

};

Поле **login**, це булеан, що відображає активну форму. Він потрібен для того, щоб вказати відповідний css-клас для div контейнера, що відтворюватимуть ефект «перетікання» з форми логіну до форми реєстрації та навпаки.

Об’єкти **signUpForm** та **signInForm** містять поля, в які записуються введені дані. Поле token міститиме jwt токен, що буде отримано від серверу у разі успішного входу або реєстрації.

Для відправки запитів на сервер я використовую Axios. Axios – це HTTP-клієнт для браузерів і node.js оснований на промісах.

**Перевірка вхідних даних**

Для перевірки вхідних даних я використовую регулярні вирази:

1. Поля типу email : ^[\w-][+@([\w-]+\.)+[\w-]{2,4}](mailto:+@([\w-%5d+\.)+%5b\w-%5d%7b2,4%7d)
2. Поля імені та прізвища : [a-zA-Zа-яА-ЯіІєЄїЇ]{1,20}
3. Поля password : ^(?=.\*\d)(?=.\*[a-zA-Z]).{8,}

У випадку введення некоректних даних дані на сервер не відправляються.

Всі стилі вказані у файлі App.css, що підключається у файлі App.js.