

Подорожники (багатосторінковий сайт)

БАЗОВА ВЕРСІЯ (MVP - minimum viable product)

Базові задачі

Верстка адаптивна, окрім мобільних пристроїв. Точки перелому:  
[Підключений нормалізатор стилів modern-normalize](#)  
Вихідний код відформатований за допомогою Prettier  
Здійснити підключення шрифтів  
Маршрутизація - App Router Next.js 15. Реалізація публічних та приватних маршрутів (захист сторінок на рівні компонентів).  
Стейт-менеджмент - використовується Redux чи Zustand, по необхідності: створення сторів, селекторів, глобального стану.  
Отримання даних: запити до бекенду, кешування, інвалідація, фонове оновлення. Для запитів до бекенда можна використовувати react quegu, якщо це потрібно.  
Індикатори завантаження - відображення ладера під час усіх активних запитів (на рівні сторінки чи окремих компонентів).  
Обробка помилок - повідомлення про помилки запитів та валідації форм (використання toast-повідомлень, статус-повідомлень на формі).  
Форми - створення форм через Formik із валідацією на основі Yup.  
Додати відображення фавікон сторінки (знаходиться в UI KIT)  
Всім інтерактивним елементам (кнопки, посилання тощо) слід додати hover-ефект згідно макету (відповідно до UI Kit).

Анімації не є обов'язковими, але їх використання може зробити взаємодію з вебсайтом більш приємною та інтуїтивно зрозумілою для користувача (наприклад можна використати бібліотеку framer motion, або [gsap](#))

Документація проєкту

У репозиторії проєкту має бути файл README, де коротко і структуровано описано:

Структура проєкту (сторінки)

- Реєстрація (route: /auth/register)

Загальні (перевикористовувані) компоненти

Layout

Загальний макет для сторінок.

Header

В Header-і знаходиться logo, навігація по сайту та кнопки авторизації/виходу.  
Logo слід реалізувати у форматі svg.  
Для мобільних пристроїв навігація по сайту має бути розміщена в меню, взаємодія з яким має бути реалізована за допомогою JS.  
Навігація по сайту має бути реалізована маршрутами, які ведуть до відповідних сторінок сайту.

Popular

Відображає список популярних історій(статей).  
Інформація про історії мандрівників повертається в результаті відповідного запиту до БД(запиту на бекенд).

Компонент містить в собі:  
Текст “Популярні історії” є заголовком секції.  
Список карток **TravellersStoriesItem** - має складатися з історій в кількості відповідно до макету.  
Кнопку "Переглянути всі" типу "button" - серверна пагінація.

Взаємодія:  
Кожен раз по кліку на кнопку "Переглянути всі" має підвантажуватись наступна порція даних з історіями до тих пір, поки не вичерпається їх перелік.  
Кількість даних в одній порції відповідає початковій кількості карток, що зображено на макеті.  
Після завантаження останньої порції даних кнопка "Переглянути всі" має бути прихована.

**TravellersStories**  
Містить в собі:  
Список карток **TravellersStoriesItem** - має складатися з історій в кількості відповідно до макету.  
Кнопку "Переглянути всі" типу "button" - серверна пагінація.

Взаємодія:  
Кожен раз по кліку на кнопку "Переглянути всі" має підвантажуватись наступна порція даних з історіями до тих пір, поки не вичерпається їх перелік.  
Кількість даних в одній порції відповідає початковій кількості карток, що зображено на макеті.  
Після завантаження останньої порції даних кнопка "Переглянути всі" має бути прихована.

**TravellersStoriesItem**  
Компонент складається з:  
зображення,  
категорію,  
назву подорожі як заголовок,  
опис,  
інформацію про автора: фото, ім’я  
дату публікації статті,  
кількість закладок - додавань до збережених  
кнопку "Переглянути статтю"  
кнопку для додавання в збережені історії - іконка.

Кнопка "Переглянути статтю" є маршрутом, який перенаправляє на сторінку "Історія" - /stories/[storyId]  
Кнопка додавання в збережені:

**Для НЕавторизованого користувача:** клік переадресовує на сторінку реєстрації (/auth/register).  
**Для авторизованого користувача:** клік виконує запит на додавання/видалення даної статті до/зі збережених статей користувача.

Під час запиту повинен відображатись індикатор завантаження(лоадер)  
Після успішного запиту значення "кількості закладок" повинно збільшитись/зменшитись на 1, а кнопка повинна змінити стилі відповідно до макету.  
Кнопка повинна мати стилі акценту, якщо стаття, що відображає дана картка на сторінці, вже додана до збережених статей авторизованого юзера.  
Після помилки в результаті запиту юзеру потрібно вивести пуш-повідомлення з описом помилки.

**TravellersList**

Під заголовком відображений перелік карточок про мандрівників, який слід реалізувати за допомогою <ul>.

Інформація про мандрівників з'являється по get-запиту на бекенд.

Під TravellersList потрібно реалізувати серверну пагінацію. Кожен раз по clickу на кнопку "Показати ще" типу "button" має додатково підвантажуватись ще три карточки з мандрівниками до тих пір, поки не вичерпається їх перелік. В такому разі кнопка "Показати ще" має бути прихована.

Кожен елемент списку містить фото, ім'я мандрівника, опис, кнопку "Переглянути профіль".

Кнопка "Переглянути профіль" є маршрутом, який перенаправляє на сторінку "Профіль".

Зображення потрібно реалізувати як контенті.

TravellerInfo

Дані про мандрівника приходять з бекенду.

MessageNoStories

Компонент містить в собі:

Footer

В Footer-і знаходиться лого, список соцмереж та напис: "© 2025 Подорожники. Усі права захищені.", перелік якірних посилань по сайту.

Logo слід реалізувати у форматі svg.

Перелік посилань на соц. мережі компанії має бути реалізовано як список елементів за допомогою <ul>, що відкриватимуть у новій вкладці відповідної соц. мережі:

Перелік навігаційних посилань, які ведуть до відповідних секцій сайту, мають бути реалізовано як список елементів за допомогою <ul>.

Сторінка "Головна" | HomePage | route: /

Компонент містить в собі компоненти:

Hero, About, Popular (загальний компонент), OurTravellers, Join	HomePage
Текст “Відкрийте світ подорожей з нами!” є <b>головним заголовком</b> сайту. Секція містить <b>описову частину</b> під заголовком. Посилання "Доеднатись" є якорем, який веде до секції-компоненту "Join".	Hero
Зображення слід реалізувати як фонове та взяте з макету. Текст “Проект, створений для тих, хто живе подорожами” є заголовком секції. Секція також містить <b>описову частину</b> праворуч від зображення.	About
Відображає інформацію про зареєстрованих мандрівників. Компонент містить в собі: Текст “Наші Мандрівники” є <b>заголовком</b> секції. Під заголовком підключений компонент <b>TravellersList</b> , список якого має складатися з чотирьох мандрівників на всіх девайсах. Під <b>TravellersList</b> потрібно реалізувати <b>пагінацію</b> . Кожен раз по click-у на кнопку "Переглянути всіх" типу "button" має додатково підвантажуватись ще три карточки з історіями до тих пір, поки не вичерпається їх перелік. В такому разі кнопка "Переглянути всіх" має бути прихована.	OurTravellers

Компонент містить в собі:  
Текст “Приєднуйтесь до нашої спільноти” є заголовком секції.  
Описову частину.  
Кнопку:  
для НЕавторизованого користувача - "Зареєструватися".  
для авторизованого користувача - "Збережені".

Join

Для авторизованого користувача: клік переадресовує на сторінку"Збережені історії" (/auth/profile).  
Для НЕавторизованого користувача: клік переадресовує на сторінку реєстрації (/auth/register).

Сторінка "Історії" | StoriesPage | route: /stories

Компонент містить в собі:  
Заголовок "Історії Мандрівників".  
Під заголовком підключений компонент TravellersStories, список якого має складатися з дев'яти історій мандрівників на десктопі, восьми - на планшеті та дев'яти - на мобільній версії.  
Під TravellersStories потрібно реалізувати пагінацію. Кожен раз по кліку на кнопку "Переглянути всі" типу "button" має додатково підвантажуватись ще три карточки з історіями до тих пір, поки не вичерпається їх перелік. В такому разі кнопка "Переглянути всі" має бути прихована.

StoriesPage

Сторінка "Історія" | StoryPage | route: /stories/[storyId]

Компонент містить в собі:  
Заголовок, наприклад "Венеція без туристів: маршрути для справжніх мандрівників", який приходить з бекенду.  
Компонент StoryDetails з деталями про історію мандрівки.  
Popular, який містить на десктопній версії три статті, на планшетній та мобільній - дві.  
Рендериться у разі, якщо запит за контентом сторінки про історію пройшов успішно.  
Завантаження даних: Під час запиту про історію повинен відображатись "лоадер".  
Компонент відображає детальний опис обраної вісторії мандрівника:  
"Автор статті": ім'я та прізвище.  
"Опубліковано": дата публікації.  
Текст з назвою країни.  
Зображення (контентне).  
Опис мандрівки.  
Блок для додавання в Збережені: заголовок "Збережіть собі історію", текст "Вона буде доступна у вашому профілі у розділі збережене" та кнопка "Зберегти".

StoryPage

StoryDetails

Взаємодія з користувачем:  
При клікн на кнопкв "Зберегти". StoryDetails оновлюється відповідною інформацією через запит на бекенд.

Сторінка "Мандрівники" | TravellersPage | route: /travellers

Компонент містить в собі:

**Заголовок** "Мандрівники"

Під заголовком підключений компонент **TravellersList**, список якого має складатися з дванадцяти мандрівників не декстопній версії, восьми - на планшетній та мобільній версіях..

TravellersPage

Під **TravellersList** потрібно реалізувати **пагінацію**. Кожен раз по кліку на кнопку "Переглянути всі" типу "button" має додатково підвантажуватись ще чотири карточки з історіями до тих пір, поки не вичерпається їх перелік. В такому разі кнопка "Переглянути всі" має бути прихована.

**Сторінка "Профіль Мандрівника Публічний" | TravelerPage | route: /travellers/[travellerId]**

Компонент містить в собі:

**TravellerInfo** містить інформацію про мандрівника з travellerId,

**Заголовок** "Історії Мандрівника",

**TravellersStories** містить список історій, які належать мандрівнику, відображеному на сторінці (з шести елементів на десктопі, чотирьох - на планшеті та мобільній версії). Даний компонент має відображатись при наявності даних про історії після запиту на бекенд.

TravellerPage

Якщо після get-запиту на бекенд дані про історії відсутні, то замість **TravellersStories** відображаємо компонент **MessageNoStories** .

**Сторінка-профіль "Збережені / Мої історії" | ProfilePage | route: /profile**

Компонент містить в собі:

**TravellerInfo** містить інформацію про мандрівника,

Список з двох **перемикачів** "Збережені історії" та "Мої історії",

**TravellersStories** містить список історій, які належать мандрівнику, відображеному на сторінці.

При активному перемикачу "Збережені історії" відображається інформація про статті (в списку **TravellersStories**), додані користувачем до "Збережених".

ProfilePage

При активному перемикачу "Мої історії" відображається інформація про статті (в списку **TravellersStories**), створені користувачем.

Взаємодія:

Після вибору іншого перемикача відбувається окремий запит на бекенд для його оновлення.

**Сторінка "Створення історії" | AddStoryPage | route: /stories/create**

Компонент містить в собі:

**Заголовок** ("Створити нову історію").

AddStoryPage

Форма з полями - **AddStoryForm**

Отримує дані в поля для вводу від користувача та відправляє запит на бекенд для збереження інформації в БД.

Форма містить в собі поля:

- Обкладинка статті - елемент що відображає обране зображення або плейсхолдер, за відсутності зображення
- Поле для підвантаження фото.
- Заголовок - текстове поле для вводу назви історії.
- Категорія - випадаючий список для вибору категорії історії.
- Текст історії - поле для введення опису історії.
- Кнопки "Зберегти" та "Відмінити".

AddStoryForm

Взаємодія з користувачем:

- Кнопка "Зберегти" - не активна до поки не будуть заповнені всі поля.
- При сабміті форми (натисканні на кнопку "Зберегти") повинна відбутись перевірка введених даних на валідність.
- Валідація полів форми відбувається за допомогою (Formik, Yup).
- Якщо введені дані НЕ валідні - виводяться повідомлення з помилками валідації біля відповідних полів згідно з UI Kit.
- Якщо введені дані валідні - відбувається запит на бекенд для створення/редагування історії.
- Після успішного запиту потрібно переадресувати користувача на сторінку створеної статті за маршрутом route: /stories/[storyId].
- При переході на вкладку "Мої історії" в профілі користувача, створена історія повинна бути в списку.
- У разі помилки запиту - форма залишається відкритою, а користувачу виводиться модальне вікно "Помилка збереження".

Сторінка "Редагування історії" | EditStoryPage | route: /stories/[storyId]/edit

Компонент аналогічний до компоненту CreateStoryPage

Компонент містить в собі:

Заголовок ("Редагувати історію").

Форма з полями - AddStoryForm отримує дані в поля для вводу від користувача та відправляє запит на бекенд для збереження інформації в БД.

EditStoryPage

Взаємодія:

- Поля форми заповнені даними відповідної статті.
- Елемент, що відображає зображення статті, повинен відображати в собі дане зображення.

Універсальне модальне вікно "Підтвердження виходу / Помилка збереження" | ConfirmModal

Універсальний компонент, реалізований як модальне вікно, що приймає пропси для налаштування контенту та його поведінки.

Компоненти:

**Заголовок** ("Ви точно хочете вийти?" / "Помилка під час збереження").

**Текст** ("Ми будемо сумувати за вами!" / "Щоб зберегти статтю вам треба увійти, якщо ще немає облікового запису зареєструйтесь").

Кнопка **закриття** ("хрестик").

**Дві кнопки** для підтвердження та відміни дії, які відправляють запит на бекенд для збереження інформації в БД..

Пропси:

*title* (string): Текст запитання (напр., "Ви точно хочете вийти?", "Помилка під час збереження").

*confirmButtonText* (string): Текст на кнопці підтвердження (напр., "Вийти", "Зареєструватись").

*cancelButtonText* (string): Текст на кнопці скасування (напр., "Відмінити", "Увійти").

*onConfirm* (function): Колбек-функція, що виконується при кліку на кнопку підтвердження.

*onCancel* (function): Колбек-функція, що виконується при кліку на кнопку скасування.

ConfirmModal

Взаємодія з користувачем:

Клік по кнопці підтвердження викликає onConfirm.

Клік по кнопці скасування, бекдропу або натискання Escape закриває модальне вікно.

Модальне вікно повинно закриватись по кліку на бекдроп (backdrop), кнопку закриття "хрестик", а також по натисканню на клавішу Escape.

Сторінка "Реєстрація / Вхід" | AuthPage | route: /auth/[authType]

Містить в собі компоненти:

RegistrationForm (відображається на маршруті **/auth/register**),

LoginForm (відображається на маршруті **/auth/login**)

Надання можливості реєстрації нового користувача.

AuthPage

Взаємодія з користувачем:

Користувач заповнює форму з полями: ім'я, email, пароль.

Користувач натискає кнопку "Зареєструватися".

Після сабміту форми відбувається перевірка введених даних на валідність.

У разі помилок валідації, вони відображаються згідно з UI Kit.

Після успішного запиту користувач авторизується і перенаправляється на сторінку "Головна".

У разі помилок в запитах виводиться пуш-повідомлення з повідомленням відповідної помилки, форма залишається заповненою.

Надання можливості входу існуючого користувача.

RegistrationForm

Взаємодія з користувачем:

Користувач вводить email та пароль.

Користувач натискає кнопку "Увійти".

Після натискання на кнопку відбувається валідація полів форми за допомогою (Formik, Yup).

У разі помилок валідації, вони відображаються згідно з UI Kit.

Після успішного запиту користувач авторизується і перенаправляється на сторінку "Головна".

У разі помилок в запитах виводиться пуш-повідомлення з повідомленням відповідної помилки, форма залишається заповненою.

LoginForm

**ДОДАТКОВІ ЗАВДАННЯ (ВИКОНУЮТЬСЯ ЗА УМОВИ, ЩО БУЛО РЕАЛІЗОВАНО ВСІ ОСНОВНІ)**

**Сторінка "Персональна інформація (редагування профілю)" | PersonalPage | route: /edit**

Додаткове завдання: Реалізувати логіку редагування профілю мандрівника

**PersonalPage**

**Сторінка "Профіль" | ProfilePage | route: /profile**

При зміні email, після основного запиту на збереження, виконувати додатковий запит, який надсилає лист для верифікації на електронну адресу, яку намагаються змінити.

**ProfileEditForm**

**Тема додатку**

Реалізувати темну тему додатку. Колір темної теми поданий в UI-kit.

## Інформація для наповнення колекцій

### BackEnd

Розгорнути сервер для розробки (підключити модулі, налаштувати CORS, написати функцію для вилову помилок тощо)

Обговорити структуру, ініціалізувати та підключити БД до проєкту

Створити swagger-документацію енд-поінтів

### /api/auth

створити публічний ендпоінт реєстрації користувача

створити публічний ендпоінт логізації користувача

написати прошарок авторизації

створити приватний ендпоінт для логаута користувача

### /api/users

створити публічний ендпоінт на отримання даних про користувачів(авторів) + пагінація

створити публічний ендпоінт на отримання даних про користувача за ID - дані користувача + список статей

створити приватний ендпоінт на отримання інформації про поточного користувача

створити приватний ендпоінт для додавання статті до збережених статей користувача

створити приватний ендпоінт для видалення статті зі збережених статей користувача

створити приватний ендпоінт для оновлення аватару користувача

створити приватний ендпоінт для оновлення даних користувача

### /api/stories

створити публічний ендпоінт для ОТРИМАННЯ історій + пагінація + фільтрація за категоріями

створити приватний ендпоінт для СТВОРЕННЯ історії

створити приватний ендпоінт для РЕДАГУВАННЯ історії

## ДОДАТКОВІ ЗАВДАННЯ (ВИКОНУЮТЬСЯ ЗА УМОВИ, ЩО БУЛО РЕАЛІЗОВАНО ВСІ ОСНОВНІ)

### /api/users

створити приватний ендпоінт для оновлення даних юзера + додати верифікацію через відправку листа на емейл, який намагаються змінити



#### auth form

name - max: 32, required,email - correct email, max: 64, required,password - min: 8, max: 128, required

#### story form

storyImage - binary, max: 2MB, required,title - max: 80, required,description - max: 2500, required,category - categoryId, required.

#### traveler form

name - max: 32, required,email - correct email, max: 64, required,avatar - binary, max: 500kB required,description - max: 150, required.