

Технічне завдання

Прототип:

(Посилання на [Figma](#) та [прототип](#) (fit to screen*))

Завдання:

Головне завдання програми - це створення продукту, яке імітує класичний Web3 проект, який включає лендинг і децентралізований веб-додаток. Лендінг буде включати різні таблиці, форми, картки і анімації, де головна мета розробника - це створення правильної верстки та адаптація всіх елементів. Веб-додаток буде включати в себе верстку, адаптацію, авторизацію та взаємодію з різними контрактами, де метою розробника буде створення коректно працюючого децентралізованого веб-додатка на всіх пристроях.

Part 1 - Landing

Дедлайн: 2,5 тижні

Загальні вимоги для розробників

Розробка інтерфейсу:

- Використання CSS змінних (значення можна знайти в UI-кіту).
- Використання семантичної верстки. Забезпечення відповідності принципу DRY (Don't Repeat Yourself).
- Створення максимально пере використовуємих компонентів для полегшення розширення та обслуговування.

Анімації:

- Для анімацій в секції Hero можна використовувати анімаційні бібліотеки або застосовувати нативний CSS.
- Посилання "Join Now" у заголовку повинно вести на форму.
- Всі посилання повинні відкриватися у новій вкладці.

Якість коду та коміти:

- Важливість чистоти та лаконічності при написанні коду.
- Забезпечення докладних назв комітів з використанням тегів (ADD, FIX, DEL, UPD) для кращої зрозумілості внесених змін. (приклад ADD | hero section, FIX | validation error)

Загальні вказівки:

- Зверніть увагу на продуктивність та оптимізацію коду.
- Забезпечення зручності та легкості розширення і підтримки.
- Ці додаткові вказівки спрямовані на поліпшення роботи з кодом та розвитку навичок розробників.

Адаптив

1. Реакція на різні роздільні здатності:
 - Перевірте, як ваша верстка виглядає на різних розширеннях екранів, включаючи, Максимальна роздільна здатність 1920, так як ширина контентної частини 1440, контент просто розташовується по центру, а мінімальна 375.
 - Впевніться, що контент центрується та налаштований для зручного перегляду та використання на різних пристроях, зокрема на планшетах та мобільних телефонах.
2. Плавні переходи та анімації:
 - Зверніть увагу на продуктивність та забезпечення плавності анімацій на різних пристроях.
3. Тестування різних браузерів та пристроїв:
 - Впевніться, що адаптивний дизайн працює на різних веб-переглядачах (Chrome, Firefox, Edge тощо) та пристроях різних виробників.
4. Ретельний аналіз на різних пристроях:
 - Здійсніть докладний аналіз всіх елементів і блоків вашого дизайну на різних розширеннях екранів, щоб переконатися, що нічого не виглядає дивно або незручно.
5. Мобільна оптимізація:
 - Зверніть увагу на швидкість завантаження сторінки на мобільних пристроях, використовуйте стиснені зображення та інші методи для забезпечення ефективності.

Форма

1. Валідація полів:
 - Забезпечте, щоб всі поля форми були обов'язковими для заповнення перед відправкою.
 - Розробіть валідацію для кожного поля, щоб переконатися, що введені дані відповідають очікуваному формату та вимогам.
2. Стабільність інтерфейсу:
 - При виникненні помилок у введенні не допустіть "перескоків" або зміщень інтерфейсу. Повідомлення про помилки мають відображатися так, щоб не порушувати загальний вигляд форми.
3. Поле для пароля з іконкою:
 - Реалізуйте додаткове завдання з інтеграцією іконки (глаза), яка дозволяє користувачеві переключати режим відображення пароля (звичайний текст або прихований).
4. Валідація після втрати фокусу:
 - Валідація полів має відбуватися після втрати фокусу (після того, як користувач перейшов з поля вводу).
 - Забезпечте, щоб повідомлення про помилку або підказки з'являлись чітко та зрозуміло після втрати фокусу.
5. Доступність:
 - Пам'ятайте про дотримання стандартів доступності, щоб інтерфейс був доступним для всіх користувачів, включаючи тих, хто використовує скрінрідери чи має фізичні обмеження.
6. Тестування на різних пристроях:
 - Впевніться, що форма здатна працювати на різних розмірах екрану та пристроях, включаючи мобільні пристрої та планшети.
7. Документація:
 - Додайте коментарі та описи до коду, якщо це необхідно для пояснення функціональності та деталей реалізації.

Загальні вимоги до роботи ПМа

1. Створення чату для своєї команди (на розсуд ПМа- Telegram, Discord, Slack ect.);
2. Створення проекту в менеджері завдань Jira та інтеграція команди;
3. Декомпозиція проекту разом з девелопером та створення беклогу;
4. Аналіз завдань, декомпозиція проекту разом з розробником та їх естімація;
5. Організація щоденних дейлі мітингів з розробниками, та сінків з тімлідами щотижня для обговорення питань;
6. Моніторинг графіків дедлайнів;

7. Взаємодія з замовниками (представниками Dexola), забезпечення прозорості процесів та звітності;
8. Створення внутрішньої документації
 - a. Створення юзер кейсів (Потрібно знайти приклади та створити документацію)
 - b. План демо (за драфтами звертатися до Олексія)
 - c. Тайм/дев репорти (за драфтами звертатися до Олексія)

Part 2 – Staking

Дедлайн: 2,5 тижні (08.09.2023 -27.09.2023)

Посилання на [Figma](#)

Загальні вимоги для розробників

Розробка інтерфейсу:

- Використання CSS змінних (значення можна знайти в UI-кіту);
- Використання семантичної верстки. Забезпечення відповідності принципу DRY (Don't Repeat Yourself);
- Створення максимально пере використовуємих компонентів для полегшення розширення та обслуговування;
- Валідація і унеможливлення виконання неправильних дій користувачем.

Якість коду та коміти:

- Важливість чистоти та лаконічності при написанні коду;
- Забезпечення докладних назв комітів з використанням тегів (ADD, FIX, DEL, UPD) для кращої зрозумілості внесених змін. (приклад ADD | hero section, FIX | validation error);
- Розділення на окремі функції по роботі з блокчейном;
- Універсалізація логіки (наприклад єдина логіка для обробки помилок);
- Пристальна увага до перевірки працездатності на мобільних девайсах.

Загальні вказівки:

- Зверніть увагу на продуктивність та оптимізацію коду;
- Забезпечення зручності та легкості розширення і підтримки;
- Ці додаткові вказівки спрямовані на поліпшення роботи з кодом та розвитку навичок розробників.

**Адаптив розробляємо за тим же принципом, як і для лендінгу.*

Форма

1. Валідація полів:
 - Забезпечте, щоб всі поля форми були обов'язковими для заповнення перед відправкою;
 - Розробіть валідацію для кожного поля, щоб переконатися, що введені дані відповідають очікуваному формату та вимогам.
2. Стабільність інтерфейсу:
 - При виникненні помилок у введенні не допустіть "перескоків" або зміщень інтерфейсу. Повідомлення про помилки мають відображатися так, щоб не порушувати загальний вигляд форми.
3. Розрахунок рейти за тиждень:
 - Розрахунок повинен відбуватись в реал-таймі, проте врахувати оптимізацію.
4. Доступність:
 - Пам'ятайте про дотримання стандартів доступності, щоб інтерфейс був доступним для всіх користувачів, включаючи тих, хто використовує скрінрідери чи має фізичні обмеження.
5. Тестування на різних пристроях:
 - Впевніться, що форма здатна працювати на різних розмірах екрану та пристроях, включаючи мобільні пристрої та планшети.
6. Документація:
 - Додайте коментарі та описи до коду, якщо це необхідно для пояснення функціональності та деталей реалізації.

Підключення смартконтрактів токена та стекінгу на Sepolia:

[*Інструкція по підключенню смартконтрактів](#)

[*Інструкція з мінтингу токенів](#)

1. Інтеграція з смартконтрактом:
 - Забезпечити інтеграцію з попередньо задеплойним смартконтрактом стекінгу на тестнеті Sepolia. Всі взаємодії з контрактом мають бути здійснені через відповідні функції;
 - Реалізувати можливість перегляду даних про стекінг, таких як баланси, відсотки винагороди тощо.
2. Робота з транзакціями:
 - Розробіть функціональність для відправлення та моніторингу транзакцій, пов'язаних зі стекінгом. Забезпечте зручний інтерфейс для користувача для ініціювання та перевірки стану транзакцій.
3. Оптимізація та продуктивність:
 - Зверніть особливу увагу на оптимізацію взаємодії з смартконтрактом для забезпечення продуктивності та швидкодії системи.
4. Документація:
 - Додайте коментарі та описи до коду, що стосуються інтеграції з смартконтрактом, для пояснення функціональності та деталей реалізації цієї частини.

Загальні вимоги до роботи ПМа

1. Вивчити базову інформацію про Clickup
2. Вивчити базову інформацію про Notion
3. Проводити тестування функціоналу продукту учасників своєї команди
4. Вивчити види естимування проекту (Three-point estimating и т.д.)
5. Вивчити типи контрактів для проекту (T&M, FP и т.д.)
6. Створити вайфрейми продукту (вивчити, що це таке, підібрати інструменти і спробувати створити вайфрейми продукту проекту)
[Figma Прототип](#)
7. Провести ресерч і зробити ексель документ, де проект буде розбитий на два майлстоуни і в ньому буде оцінена робота за часом для кожного учасника команди (дизайнер, блокчейн розробник і т.д.).

8. Також слід побудувати діаграму Ганта, де будуть позначені часові відрізки, коли кожен з учасників береться за роботу та закінчує її. (одна в ClickUp+)
9. Зробити презентацію на 27.09 робіт

Інформація для вивчення

[What is Web 3.0?](#)

[The Non-Fungible Token Bible](#)

[What is a DAO in Crypto?](#)

[Intro to Eth2 & Staking for Beginners](#)

[Intro to Web3.js](#)

[ERC20 Token Tutorial](#)

[Ethereum & ERC-20 tokens on Trezor](#)