

Критерії приймання

Привіт, студенте! На даному етапі навчання ти будеш працювати разом з іншими учасниками твого потоку над фінальним командним проєктом з блоків `React + Node.js`. Основні завдання, які стоятимуть перед командою:

- У заданий термін командою реалізувати фінальний проєкт з блоків `React+Node.js`
- Отримати практичний досвід роботи у команді
- Взаємодіяти в команді, в тому числі при роботі з `Git`
- Розвинути навичку розгалуження та злиття проєкту
- Навчитися вирішувати конфлікти при злитті гілок
- Отримати досвід планування та постановки завдань
- Отримати досвід роботи з `Trello`
- Отримати досвід на позиції `Team Lead` команди (додатково, за бажанням)
- Отримати досвід на позиції `Scrum Master` команди (додатково, за бажанням)
- Отримати досвід презентації проєктів

Критерії допуску роботи

1. Реалізована семантична верстка з трьома переломами (для мобільних девайсів, планшетів та десктопу) і відповідає технічному завданню та дизайн-макету.
2. Підключений `modern-normalize`.
3. Підключено шрифти.
4. Всі статичні зображення попередньо оптимізовані, в т.ч. під `retina`-екрани.
5. Оптимізовано завантаження статичних зображень.
6. Додано відображення фавікон сторінки.
7. Проєкт пройшов оцінювання на <https://pagespeed.web.dev/> і кожен показник складає не менш ніж 70%.
8. Консоль розробника не видає помилок чи `console.log`'ів.
9. У назвах файлів немає великих літер і пробілів, лише літери англійського алфавіту.
10. Всі змінні іменовані по `camelCase`'у, класи по `PascalCase`'у, константи по `UPPER_SNAKE_CASE`'у.
11. Проєкт розгорнуто через `create-react-app` (у разі якщо не був використаний шаблон `react-homework-template`)
12. Усі сторінки знаходяться у папці `pages`.
13. Усі компоненти знаходяться у папці `components`.
14. Все, що пов'язано з `redux`, знаходиться в папці `redux`.
15. Підключення раутів реалізовано через `lazy load`.
16. Вся секретна інформація (якщо така є: ключі, паролі та інше) захована через змінні оточення в файлі `.env`. Також є файл `.env.template`, де вказано, які змінні потрібно ініціалізувати для запуску проєкту після клонування.
17. Всі робочі гілки окрім: `main`, `gh-pages`(якщо така є), `dev`(якщо така є) - видалені.

18. Реалізовано увесь обов'язковий функціонал, передбачений технічним завданням з Front End'а та Back End'а.
19. Оформлено документацію ендпоінтів Back End'а за допомогою Swagger.
20. Фінальна версія Front End'а задеплойена на GitHub Pages, Back End'а - на Render або його аналоги (за бажанням, студенти також можуть самостійно задеплойити свій проєкт на якийсь орендований сервер).
21. Ментори з Front End'а та Back End'а провели код-рев'ю фінальної версії проєкту.

Щоб здати ДЗ, необхідно прикріпити:

- посилання на pull-requests зі своїм кодом (pull-requests мають бути прийняті та залиті в основну гілку репозиторію командного проєкту)
- посилання на Swagger-документацію
- посилання на сторінку GitHub Pages