Требования

Клиентская часть на react + typescript

Проект должен быть расположен в репозитории (один репозиторий и на клиент и на сервер)

Каждая группа реализовывает один проект в рамках группы(один проект на всех)

Тематика проекта любая

В рамках работы над проектом допустимо разделение команды по ролям

Групповой проект ведется в одном едином репозитории

Допустимые роли на проект:

Бекенд

Фронтенд (обязательно адаптивная вёрстка)

Дизайнер1ч

Аналитик 1ч

Тимлид 1ч отвечает полностью за проект

Тестировщик

Любой проект начинается с концепции(цель и суть проекта, описание основого функционала, список платформ на которых он исполняется, основная фишка приложения, краткое описание лора)

На основании концепции аналитик пишет требования, либо тз

Описывается детально требуемый функционал, так чтобы было понятно и заказчику, и исполнителю

Инструменты и организация работы

Вся разработка и все задачи должны вестись в программе Mirror

Там должно быть:

Название приложения

Состав команды и записанными ролями и с именем учетки в гитхабе

Должна быть прописана ссылка на репозиторий с проектом, ссылка на дизайны в фигме

Доска разделённая на колонки:

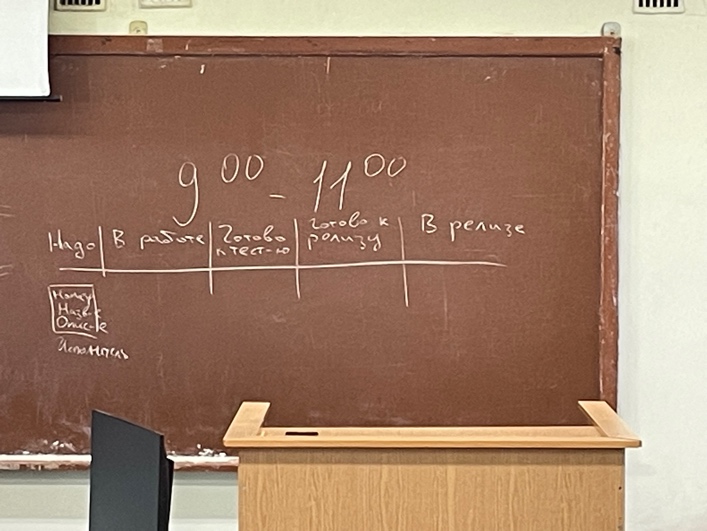
Надо делать

В работе

Можно тестировать

Готово к релизу

В релизе



Алгоритм работы – единственный способ организации работы в команде

Каждая задача выполняется в репозитории в отдельной ветке которая клонироутеся от ветки dev (основное название проекта-номер-название ветки), после того как задача сделана разработчик спуливает все последние изменения из dev и разрешает конфликты которые могут быть, после этого разработчик перетаскивает задачу к готово к тестированию, тестировщик переключается в эту ветку тестирует задачу, находит баги, оформляет баги следующими задачами, когда все баги успешно исправлены тестировщик или разработчик перетаскивает к готово к релизу и оформляет пул реквест на тимлида

В проекте две основные ветки main и dev

В ветке dev в каждый момент времени лежит рабочая версия приложения, в ветку dev пушить может только тимлид(не пушит, а разрешает pull-request)

Работа с репозиторием

Git checkout -b develop создать ветку

git checkout develop переключиться на ветку

git pull origin develop. Спулить изменения с девелоп

git checkout -b BDSM-123-pain создать свою ветку

git add .

git commit -m «комментарий».

Git pull origin develop

Решить конфликты

Git push origin BDSM-123-pain

Аналитик

Концепция требования api описание дизайна и тп должны лежать в этом же репозитории в файлах md

К следующей лекции написать сосавы команд и списком ролей, ссылку на доску и репозиторий, концепция

Поставить программу open server