

Требования к проекту

GitHub

- Создать Git-репозиторий перед началом разработки
- Завести README файл и записать туда техническое задание
- Создавать ветки для последовательной разработки
- Во время разработки отдельных частей сайта работать в отдельных ветках
- При создании, например, регистрации на сайте заводить отдельную ветку на GitHub и весь процесс разработки делать коммиты.
- При завершении разработки вы делаете Pull Request на добавление вашего кода в основную ветку.
- Менторы будут проверять написанный код, его внешний вид, соблюдение стандартов, и если все сделано как нужно, будут делать Merge вашей ветки в основную.
- В случае, если ваш код написан неверно, не соответствует требованиям, вам будет отправлен issues с описанием проблемы для дальнейшего исправления. После исправлений всех указанных проблем, нужно будет снова сделать Pull Request

Внедрена система дедлайнов. 5 дней - 1 дедлайн

Начало в понедельник (утро). Окончание - пятница (вечер).

До пятницы вам нужно будет сделать Pull Request. В течении 2 дней менторы проверяют ваш Pull Request. Если все правильно, они делают Merge (Добавляет ваш код в основную ветку). В ином случае Pull Request закрывается, отправляется Issue с описанием проблемы, после устранения которой вы должны снова сделать Pull Request

Разберем конкретный пример.

Рекомендую для разработки использовать PyCharm или VSCode, объединить их с GitHub. [Инструкция по настройке](#)

Например, Вы решили добавить на сайт регистрацию. Для этого вы создаете отдельную ветку в Github с соответствующим названием. После этого начинаете разработку, делая постоянные коммиты. Коммиты нужно делать постоянно для более гибкого контроля и отслеживания ошибок.

По окончании создания регистрации нужно сделать Pull Request.

Требования к проекту

Trello

Для контроля задач и сроков, а также для визуализации процесса разработки и повышения соревновательного духа, мы будем использовать Канбан доску «Trello».

Необходимо зарегистрироваться и отправить имейл в ЛС Дмитрию Чухнину для получения приглашения в команду

Из чего состоит доска:

Команды:

У каждой команды будет свой цвет и соответственно задачи для них

Front-end:

Тут задача висит до тех пор, пока разработка фичи не завершена. После завершения она передвигается в столбец для тестов.

Или, если архитектура неверна или неточна — задача будет возвращена обратно в столбец.

Back-end:

Тут задача висит до тех пор, пока разработка фичи не завершена. После завершения она передвигается в столбец для тестов.

Или, если архитектура неверна или неточна — задача будет возвращена обратно в столбец.

Тестирование:

В этом столбце задача находится, пока она тестируется менторами, после того, как вы завершили задачу, необходимо будет передвинуть ее в этот столбец. Если найдены ошибки — возвращается в Разработку. Если нет — передвигается дальше.

Деплоймент:

Будем мержить ветки, если задача и на фронте и на бэке готова

Готово:

Сюда стикер попадает только тогда, когда все работы по задаче закончены полностью.

Весь Канбан можно описать всего тремя основными правилами:

Требования к проекту

1. Визуализируйте производство

— Разделите работу на задачи, каждую задачу напишите на карточке и поместите на стену или доску.

— Используйте названные столбцы, чтобы показать положение задачи в производстве.

2. **Ограничивайте WIP** (work in progress или работу, выполняемую одновременно) **на каждом** этапе производства.

3. **Измеряйте время цикла** (среднее время на выполнение одной задачи) и **оптимизируйте постоянно процесс**, чтобы уменьшить это время.