

2024

1. Место и сроки проведения практики:

Наименование организации: Кафедра 806

Сроки проведения практики

дата начала практики: 28.06.2024

дата окончания практики: 11.07.2024

2. Инструктаж по технике безопасности:

_____/ Крылов С.С. / 28 июня 2024г.
подпись проводившего *расшифровка подписи* *дата проведения*

3. Индивидуальное задание обучающегося:

Разработать frontend для микросервиса
«WareHouseManagement»

4. План выполнения индивидуального задания обучающегося:

№ п/п	Место проведения	Тема	Период выполнения
1	Кафедра 806	Инструктаж.	28.06.2024-28.06.2024
2	Кафедра 806	Разработка дизайна	28.06.2024-29.06.2024
3	Кафедра 806	Проектирование архитектуры проекта	29.06.2024-01.07.2024
4	Кафедра 806	Инициализация инфраструктуры проекта	02.07.2024-03.07.2024
5	Кафедра 806	Верстка страниц и компонентов	04.07.2024-05.07.2024
6	Кафедра 806	Интеграция API сервера с веб представлением	06.07.2024-07.07.2024
7	Кафедра 806	Тестирование	08.07.2024-09.07.2024
8	Кафедра 806	Оформление отчета. Подведение итогов.	10.07.2024-11.07.2024

Утверждаю

_____/ Крылов С.С. / 28 июня 2024г.
подпись руководителя от МАИ *расшифровка подписи* *дата утверждения*

_____/ Крылов С.С. / 28 июня 2024г..
подпись руководителя от *расшифровка подписи* *дата утверждения*
организации/предприятия

Ознакомлен

_____/ Терентьев М.А. / 28 июня 2024г.
подпись обучающегося *расшифровка подписи* *дата ознакомления*

5. Отзыв руководителя практики от организации/предприятия:

Обучающийся группы М8О-101Б-23 Терентьев Михаил Андреевич проходил практику на кафедре 806 МАИ. Был разработан frontend для проекта «WareHouseManagement». За время прохождения практики, практикант показал необходимый уровень развития практических навыков и компетенций в процессе выполнения индивидуального задания. Рекомендуемая оценка «отлично». Материалы, изложенные в отчёте обучающегося, полностью соответствуют индивидуальному заданию.

*подпись руководителя от
организации/предприятия*

/ Крылов С.С. /
расшифровка подписи

11 июля 2024 г.
дата

6. Отчет обучающего по практике:

6.1. Проектирование архитектуры проекта

Для эффективного управления состоянием приложения и обработки получаемых данных, особенно в крупных проектах, часто используется библиотека Redux. Redux предоставляет централизованное хранилище (store) для всего состояния приложения, что упрощает управление данными и их синхронизацию между различными компонентами. Для каждой группы обработчиков были созданы api-компоненты, управляющие запросами и slice-компоненты, управляющие состоянием.

6.2. Инициализация инфраструктуры проекта .

Проект состоит из нескольких отдельных групп обработчиков для страниц – warehouse, chamber, floor, user, customer, inward, outward. Для создания современного и отзывчивого пользовательского интерфейса была выбрана библиотека Tailwind CSS. Tailwind предоставляет набор утилитных классов, которые позволяют быстро и эффективно создавать адаптивные дизайны без необходимости писать CSS с нуля.

Кроме того, для ускорения разработки и обеспечения единообразия компонентов была использована коллекция компонентов shadcn/ui. Эта библиотека предлагает готовые, стилизованные компоненты, которые можно легко интегрировать в проект, что значительно упрощает процесс разработки интерфейса.

Для тестирования функций сервера и обеспечения взаимодействия с API была установлена библиотека Orval. Orval автоматически генерирует типы и заглушки на основе OpenAPI (Swagger) спецификаций, что упрощает интеграцию API сервера в проект. Это позволяет разработчикам быстро и безопасно работать с API, обеспечивая высокую степень согласованности и надежности.

6.3. Верстка страниц и компонент

Были сверстаны страницы, соответствующие каждой группе обработчиков и компоненты для управления формами обработки:

- page – основная страница,
- [Product]List – компонента, генерирующая список элементов, полученных с запроса, где Product – элементы, соответствующие группе warehouse, chamber, floor и т.д.,
- Activate[Product]Popover – компонент, позволяющая сменить статус элемента,
- Add[Product]Popover – компонент, позволяющий добавить элемент,
- Delete[Product]Popover – компонент, позволяющий удалить элемент,

- Edit[Product]Popover – компонент, позволяющий редактировать элемент,
- Find[Product]Popover – компонент, позволяющий найти элемент.

Кроме того, были созданы компоненты:

- Header, Navmenu – шапка сайта с меню навигации,
- ActivateForm, AddForm, DeleteForm, EditForm, FindForm – компоненты, генерирующие соответствующую форму по входящим в нее полям.

6.3. Интеграция API сервера с веб представлением.

Для обеспечения качественной работы веб-приложения, особенно в связке с его серверной частью, крайне важно проводить комплексные тесты. Одним из эффективных инструментов для этого является библиотека MSW (Mock Service Worker). MSW позволяет имитировать HTTP-запросы и ответы, что значительно упрощает процесс тестирования и позволяет избежать зависимости от реального сервера.

Gitlab с решением всех задач: <https://gitlab.mai.ru/MATerentev/ware-house>

подпись обучающегося

/ Терентьев М.А. /
расшифровка подписи

11 июля 2024 г.
дата