



# Веб-приложение для учета посещаемости и успеваемости студентов колледжа

# Обзор Проекта и Основная Цель

Представляем инновационное веб-приложение, разработанное для оптимизации и автоматизации процессов учета посещаемости и успеваемости студентов колледжа. Этот проект создан как Single Page Application (SPA), обеспечивая мгновенную навигацию без перезагрузки страниц.

## Автоматизация учета

Эффективный учет посещаемости и оценок.

## Удобный интерфейс

Интуитивно понятный дизайн для преподавателей и администраторов.

## Быстрая навигация

SPA архитектура для бесшовного пользовательского опыта.



## Основные Возможности Системы

Приложение предлагает широкий спектр функций, адаптированных под различные категории пользователей.

1

### Для всех пользователей

- Просмотр списка студентов (с фильтрацией и поиском)
- Просмотр записей о посещаемости по датам и предметам
- Просмотр успеваемости и статистики по оценкам

2

### Для Преподавателей

- Редактирование и массовая отметка посещаемости
- Управление оценками (добавление, редактирование, удаление)

3

### Для Администраторов

- Полный CRUD для студентов, групп и предметов
- Деактивация записей вместо физического удаления

# Технологический стек: Frontend

Наше приложение построено на современных и надежных технологиях, обеспечивающих высокую производительность и стабильность.



## React 19.2.0

Ведущая библиотека для создания интерактивных пользовательских интерфейсов.



## TypeScript 5.9.3

Типизированный JavaScript для повышения надежности и поддерживаемости кода.



## Vite (rolldown-vite 7.3.0)

Быстрый инструмент сборки нового поколения для моментальной разработки.



## MobX 6.15.0

Простая и масштабируемая библиотека для управления состоянием приложения.



## SCSS/SASS 1.97.1

Мощный препроцессор CSS для модульной и гибкой стилизации.



## Firebase 12.7.0

Гибкая база данных в реальном времени для быстрого хранения и синхронизации данных.

# Архитектура Проекта: SRC-Папка

Структура проекта четко организована для облегчения разработки и масштабирования. SRC-папка содержит все ключевые модули приложения.



# Система Ролей и Прав Доступа

Приложение имеет гибкую систему ролей, которая обеспечивает разграничение доступа к функционалу на основе учетных данных пользователя.



## Гость (user)

Просмотр списков студентов, посещаемости и успеваемости без возможности редактирования.



## Преподаватель (teacher)

Дополнительно к гостю: редактирование посещаемости и управление оценками.



## Администратор (admin)

Полный контроль: все права преподавателя плюс управление студентами, группами и предметами (CRUD операции).

Авторизация пользователя определяет доступные ему функции, обеспечивая безопасность и порядок.



# Структура Данных: Ключевые Сущности

Для обеспечения целостности и эффективного хранения информации, проект оперирует пятью основными сущностями.



## Студент

id, имя, фамилия, отчество, группа, дата рождения, email, телефон, статус активности.



## Группа

id, название, описание, год обучения, статус активности.



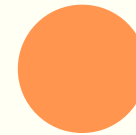
## Предмет

id, название, описание, код, статус активности.



## Посещаемость

id, студент, предмет, дата, статус (присутствовал/отсутствовал/опоздал/уважительная причина), примечания.



## Оценка

id, студент, предмет, значение, тип (экзамен, зачет), дата, примечания.

# Руководство по Запуску Проекта

Для успешного запуска приложения следуйте простым шагам, начиная с установки необходимых зависимостей.

01

## Предварительные требования

Убедитесь, что установлены Node.js (рекомендуется v18+) и npm.

02

## Установка зависимостей

```
npm install
```

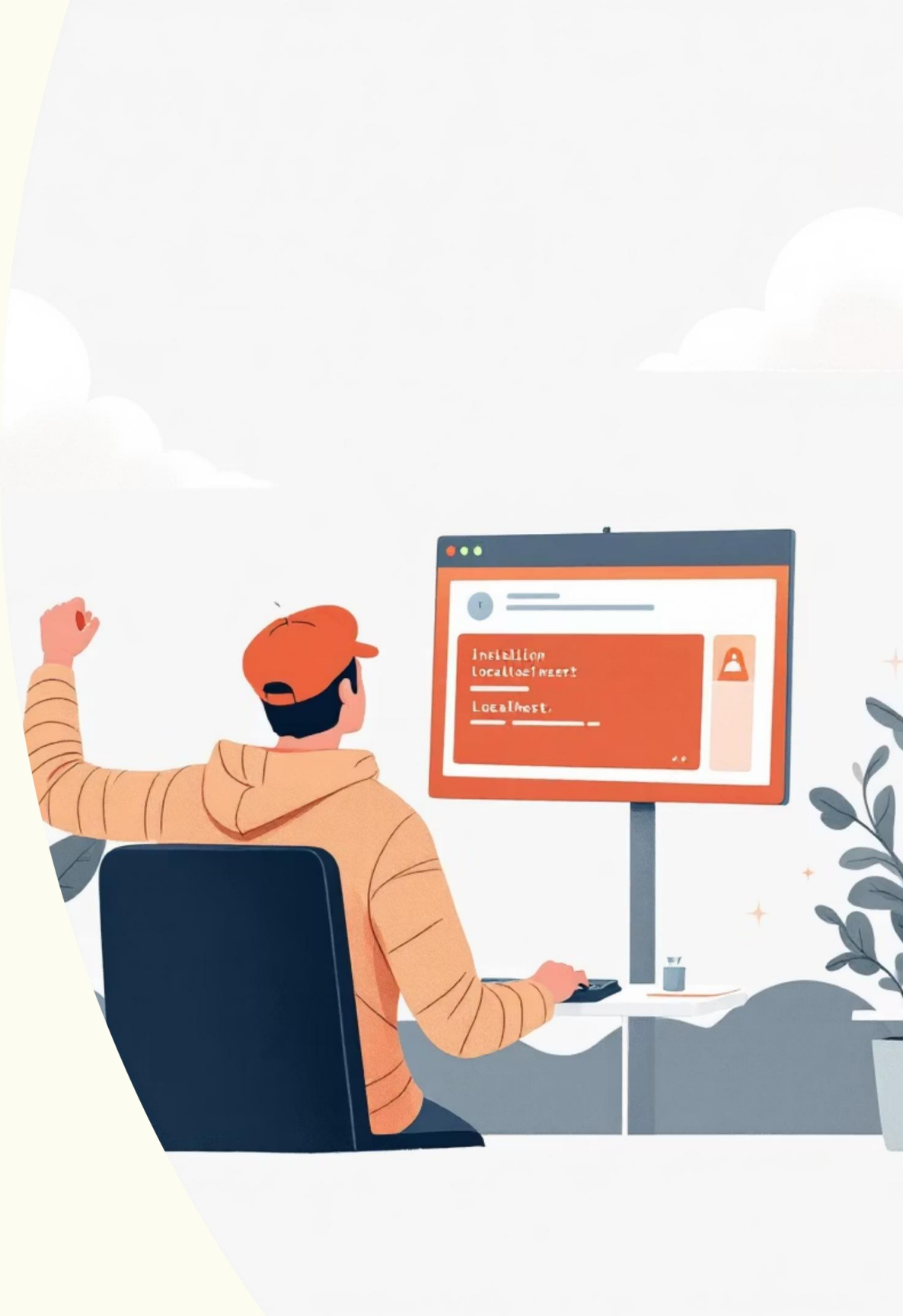
Выполните эту команду в корневой папке проекта.

03

## Запуск development сервера

```
npm run dev
```

Приложение будет доступно по адресу:  
[http://localhost:5173/<НАЗВАНИЕ\\_ПРОЕКТА>/](http://localhost:5173/<НАЗВАНИЕ_ПРОЕКТА>/)





# Работа с Проектом: Основные Принципы

Для тех, кто хочет глубже погрузиться в код или кастомизировать проект, мы предоставляем краткое руководство по навигации, работе с данными и авторизацией.

## Навигация

Использует кастомную систему через **NavigationStore**. Для перехода используйте `navigationStore.navigate('page-id')`. Права доступа проверяются автоматически.

## Авторизация

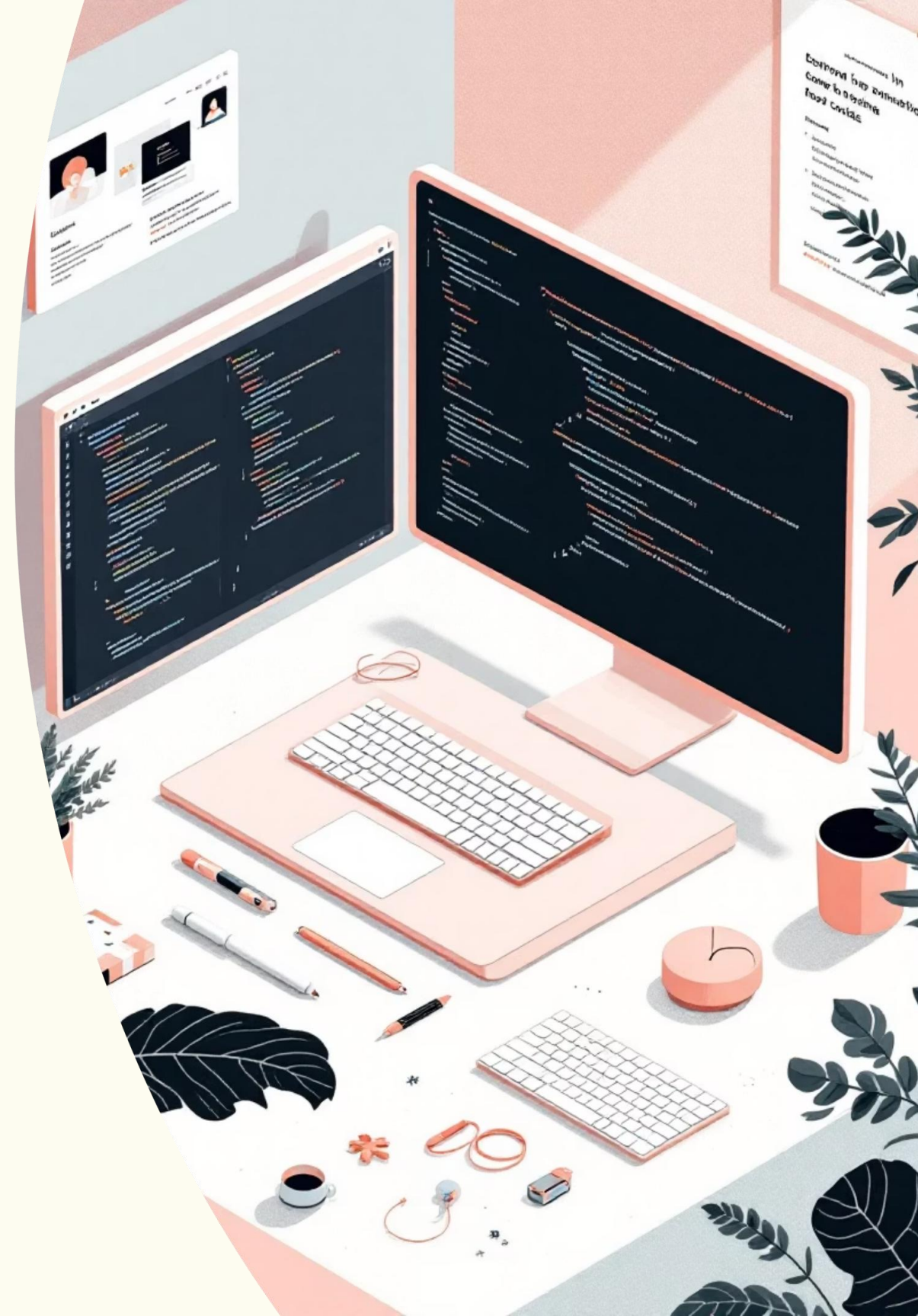
Управляется через **AuthStore**. Проверяйте роли (`authStore.isAdmin`) и права (`authStore.canEditAttendance`). Доступны методы для входа (`authStore.openLoginModal()`) и выхода (`authStore.logout()`).

## Работа с данными

Все CRUD операции выполняются через **DataStore**. Методы начинаются с `load`, `create`, `update` и `delete`. Поддерживается фильтрация данных.

## Стилизация

Проект использует модульный подход SCSS. Глобальные стили находятся в `src/styles/`, модульные — в файлах `.module.scss` компонентов. Цветовую палитру можно изменить в `globalColors.scss`.



# Заключение: Будущее Управление Колледжем

Данное веб-приложение — это значительный шаг к созданию более эффективной и прозрачной системы управления образовательным процессом в колледже.

## Повышенная Эффективность

Сокращение ручной работы и минимизация ошибок.



## Прозрачность

Легкий доступ к данным для всех уполномоченных пользователей.



## Масштабируемость

Возможность дальнейшего расширения функционала.



## Инновационное Решение

Современные технологии для лучшего образования.

