[redacted]

[redacted]

[redacted]

[redacted]
[redacted]

[redacted]

[redacted]

[redacted]

□

□□□□ □ □□□□□□ □□□□□ □□□□□	
□□□□ □ □□□□□□ □□□□□□ □□□□□□□	
□□□□□□ □□□□□	[□□□□□□ □□□□□]
□□□□□ □□□□□□ □□ □□□□□	
□□□□□ □□□□□□ □□ □□□□	
□□□□□□ □□□□□□	□□□□□□□□ □□ □□□□□□ □ □□□□.□□□□□ □□□□□□□□ □□□□□□□□□□□□□ □□□□□□ □□□□ □□□□ □□
□□□□□□ □□□□□□□□□□	[□□□□□□ □□□□□□□□□□]
□□□□ □ □□□□ □□□□□□□□□□ □□□□□□□□	[□□□□ □ □□□□ □□□□□□□□□□ □□□□□□□□]

□□□□ □□□□□ □□□□□□ □□□□□□ □□□□□□□□□□

□□□□□ □ □□□□□□□

□□□□□ □□ □□□□□□□ □□□□□ □□ □□□□ □□□□□ □□□□□□ □□□□□□ □□ □□□□□□□□□□□□ □□□□ □□□□□□ □□ □□□□□ □□□□□□□□ □□ □□□□□ □□□□□□□□ □□□□□□□□ □□□□ □□□□ □□□□□ □□□□ □□ □□ □□ □□□□□□ □□□□□□□□ □□□□□□□□ □□□□ □□□□ □□□□□ □□□□ □□ □□□□□□ □□□□□□□□ □□□□□□□□ □□□□ □□□□ □□□□□ □□□□ □□ □□□□□□ □□□□□□□□ □□□□□□ □□□□ □□□□ □□ □□□□□□ □ □□□□□□ □□□□□□□□ □□□□□□ □□□□ □□□□ □□

[illegible]

[illegible]

□

□□□□□□ □□□□□□□

□□□□□□ □□□□□□□

□□□□□

□□□□□ □□□□ □□□□□ □□□□□□□□□□□□□□□□ □ □□□□□ □□ □□□□□ □□□□□ □□□□□□□ □□ □□□□□ □□ □□□□□ □□□□□ □□□□□□□□□□□□□□□□ □□ □□□□□□□□ □□□□□□□□□□ □□ □□□□□ □□□□□ □□□□□ □□ □□□□□□□□ □□□□□□□□□□ □□ □□□□□□□□ □□ □□□□ □□ □□□□ □□□□□ □□□□ □□ □□□□□ □□□□□□□□□□ □ □□□□□□□□ □□□□□□□ □□ □□ □□□□ □□□□□□□□□□ □□□□□□□□□□ □□ □□□□□ □□□□□ □□□□ □□□□ □□□□□ □□□□□□□□□□□□□□□□ □□ □□□□□□□□ □□□□□□□□□□ □□□□□ □□□□□□□□□□□□□□□□

5 Planning a Software Project

5.1 Project Scope

Before we can start planning a software project, we must first understand what the project is supposed to do. We must understand the problem that the software is supposed to solve, and we must understand what the software is supposed to do. We must understand what the software is supposed to do in order to be able to plan the project. We must understand what the software is supposed to do in order to be able to estimate how much work it will be to develop the software. We must understand what the software is supposed to do in order to be able to estimate how much it will cost to develop the software (both in terms of money and in terms of time).

5.2 How to Plan a Software Project

In this section we will plan the development of the RecipeShare software. We will not be planning the development of the entire software system, but we will plan the development of a small part of the system. We will plan the development of the part of the system that is responsible for storing recipes in a database (and retrieving them from the database). We will not be planning the development of the part of the system that is responsible for displaying recipes on the screen. We will not be planning the development of the part of the system that is responsible for allowing the user to enter new recipes into the system.

5.3 Requirements Analysis

The first step in planning the development of the RecipeShare software is to understand what the software is supposed to do. We will do this by:

- Talking to the users of the software
- Reading existing documentation about the system
- Looking at existing software systems that are similar to the one we are going to develop
- Talking to the developers who are going to develop the software
- Talking to other people who have developed similar software systems in the past

Глава 3: Создание и настройка базы данных

Введение

RecipeShare использует базу данных для хранения информации о пользователях, рецептах и комментариях. В этой главе мы настроим базу данных и создадим необходимые таблицы.

Установка и настройка базы данных

Установка MySQL

Для работы с базой данных мы будем использовать MySQL. Установите MySQL Server и MySQL Workbench с официального сайта.

Создание базы данных и пользователя

Создайте базу данных и пользователя для приложения:

```
CREATE DATABASE recipeshare;
CREATE USER 'recipeshare_user'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON recipeshare.* TO 'recipeshare_user'@'localhost';
FLUSH PRIVILEGES;
```

Настройка подключения к базе данных

Настройте подключение к базе данных в файле конфигурации приложения.

Создание таблиц

Таблица пользователей

Создайте таблицу для хранения информации о пользователях:

```
CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL UNIQUE,
    password_hash VARCHAR(255) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Таблица рецептов

Создайте таблицу для хранения рецептов:

```
CREATE TABLE recipes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT NOT NULL,
    title VARCHAR(100) NOT NULL,
    description TEXT,
    ingredients TEXT NOT NULL,
    instructions TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

Таблица комментариев

Создайте таблицу для хранения комментариев к рецептам:

```
CREATE TABLE comments (
    id INT AUTO_INCREMENT PRIMARY KEY,
    recipe_id INT NOT NULL,
    user_id INT NOT NULL,
    comment TEXT NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (recipe_id) REFERENCES recipes(id),
    FOREIGN KEY (user_id) REFERENCES users(id)
);
```

Заключение

В этой главе мы настроили базу данных и создали необходимые таблицы для приложения RecipeShare. В следующей главе мы создадим API для взаимодействия с базой данных.

[illegible]

[illegible]

[illegible] RecipeShare [illegible].

[illegible]

[illegible] RecipeShare [illegible]:

- **[illegible]:** [illegible] [illegible].
- **[illegible]:** [illegible] [illegible].

[illegible]

- [illegible]
- [illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible]

[illegible]:

- **Users:** [illegible] (email).
- **Recipes:** [illegible].
- **Comments:** [illegible].

[illegible]

API] [illegible]

2.2 Структура проекта

2.2.1 Структура в Angular CLI

Angular CLI создает стандартную структуру проекта, которая организует код по функциональным модулям и компонентам. Основные директории:

- **Components:** Здесь находятся компоненты, которые можно использовать в разных частях приложения.
- **Pages:** Содержит компоненты, которые представляют собой отдельные страницы приложения.
- **Assets:** Хранит статические файлы, такие как изображения и CSS.

2.3 Реализация

[Здесь реализация]

[□□□ □□□□□ □□□□□]

□•□ □□□□□□ □□□□□□ □□ □□□□□

□□□□□□ □□ □□□□□ □□□□□•
□□□□□ □□□□□ □□ □□□□□ □□□□□• □□ □□□ □□□□□□ □□□□□ □□□□□□ □□□□□□ □□□□ □
□□□□ □□ □□□□□ □□□□ □□□• □□□□□•□□□□ □□□□□□□□□□□□ □□ □□□□□□□□□ □□□□□□□
□□□□□□□ □□ □□□•□□□□ □ □□□□□□ □□□□□ □□□□□□ □□□□□□□□□□□ □□□□□□ □□ □□□□

□•□ □□□□□□□ □□□ □□□□□□□ □□ □□□□□□ □□□□□

□ □□□□□□□ □ □□□□□□□□□

Abstract

discover share to users allow to designed application web a is RecipeShare with Node.js end front the for React using Built recipes. cooking manage and user- a provides it database the as MongoDB and end back the for Express like features supports system The enthusiasts. culinary for platform friendly engagement. user for analytics and categorization sharing recipe Recipe MongoDB Express Node.js React Application Web Keywords: Sharing.

□□□ □□□□□□□□ □□□□□□

- □□□□□□ □□□□□□□□: https://nodejs.org/en/docs/
- □□□□□□ □□□□□□□: https://expressjs.com/
- □□□□□□ □□□□□□□: https://docs.mongodb.com/
- □□□□□□ □□□□□□□: https://reactjs.org/docs/