Homyz - Technical Design Document

Table of Contents

Introduction and System Overview

Introduction

HomyZ is a modern real estate web application built using the MERN (MongoDB, Express.js, React, Node.js) stack. The platform allows users to list, search, and explore properties seamlessly. It offers robust user authentication, responsive design, and efficient database management to ensure a smooth user experience.

System Overview

HomyZ consists of a frontend built with React.js for a responsive user interface and a backend powered by Node.js with Express.js for handling API requests. MongoDB serves as the database for storing user and property data. The application supports user authentication and CRUD operations for properties.

Key Features

- **Property Listing**: Users can list properties with details such as images, descriptions, price, and location
- **Search Functionality**: Users can search and filter properties based on criteria like price range, location, and type
- **Image Upload**: Properties can include multiple images, stored in a dedicated uploads directory

- **Responsive Design**: Optimized for both desktop and mobile devices

Technologies Used

- **Frontend**
  - React.js: For building a dynamic and responsive user interface
  - CSS: For styling components and pages
- **Backend**
  - Node.js: JavaScript runtime for server-side scripting
  - Express.js: Web application framework for API creation

- **DataBase**
  - MongoDB: NoSQL database for storing user and property data
  - Mongoose: ORM for schema definition and database interaction

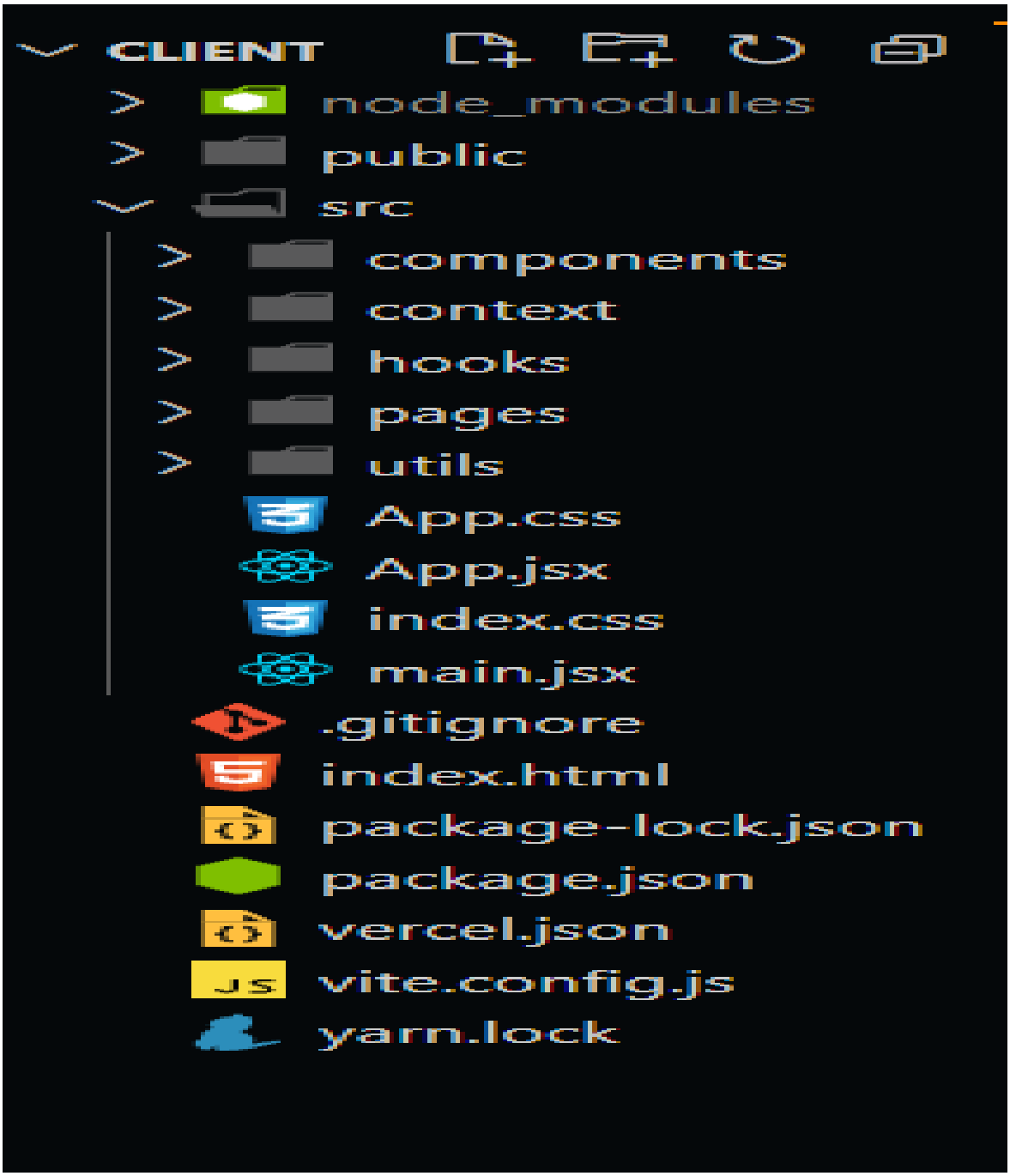
High Level Architecture

- **Client Layer:** React.js-based frontend that interacts with the backend APIs and presents data to users
- **Server Layer:** Node.js backend with Express.js to handle API routes, middleware, and business logic
- **Database Layer:** MongoDB database accessed through Mongoose ORM
- **Authentication:** Secure login and session handling using JWT
- **Static File Handling:** Uploads directory for storing property images

File Structure

Backend

Frontend


CLIENT
node_modules
public
src
components
context
hooks
pages
utils
App.css
App.jsx
index.css
main.jsx
.gitignore
index.html
package-lock.json
package.json
vercel.json
vite.config.js
yarn.lock

Database Schema

User Model

Fields:

- E - mail
- **password**

Add Property

Fields:

- **name**
- **address**
- **Details**
- **Price**
- **Image**