SYNOPSIS

Report on

Chat application

by

Mohd Nazim: 2200290140094
Pawan Sharma: 2200290140107
Pranav Chaudhary:2200290140112

Session:2023-2024 (IV Semester)

Under the supervision of

Prof. Shweta Singh

KIET Group of Institutions, Delhi-NCR, Ghaziabad

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR,
GHAZIABAD-201206**

(APRIL- 2024)

ABSTRACT

The "ChatApplication" project is an Angular-based chat application. It uses CLIversion 16.0.0 and includes features such as user authentication with credentials stored in `db.json`. To set up, clone the repository, install dependencies via `npm install`, and run the server using `npx json-server --watch db.json`. The application can be accessed locally at `http://localhost:4200/`. The project also supports code scaffolding, building, unit tests with Karma, and end-to-end tests.

Angular CLI (Command Line Interface) is a tool that helps developers initialize, develop, scaffold, and maintain Angular applications. It simplifies the setup process, manages dependencies, and provides commands for building, testing, and deploying Angular projects. With Angular CLI, developers can quickly generate components, services, modules, and other Angular artifacts, ensuring a consistent project structure and development workflow. It also offers built-in support for best practices like TypeScript, Webpack, and Angular Material.

Node.js is a crucial part of the ChatApp-Angular project, serving as the runtime environment for the server-side logic. It allows for the creation of a lightweight and efficient backend using JavaScript. In this chat application, Node.js, along with tools like `json-server`, is used to manage and serve the application's data, handle API requests, and ensure real-time data updates. This setup facilitates seamless communication between the client (Angular) and the server, ensuring a responsive and interactive user experience.

TABLE OF CONTENTS

Page Number

INTRODUCTION

OVERVIEW

The "ChatApplication" project is an Mern-based chat application. It uses Angular CLI version 16.0.0 and includes features such as user authentication with credentials stored in `db.json`. To set up, clone the repository, install dependencies via `npm install`, and run the server using `npx json-server --watch db.json`. The application can be accessed locally at `http://localhost:4200/`. The project also supports code scaffolding, building, unit tests with Karma, and end-to-end tests.

Angular CLI (Command Line Interface) is a tool that helps developers initialize, develop, scaffold, and maintain Angular applications. It simplifies the setup process, manages dependencies, and provides commands for building, testing, and deploying Angular projects. With Angular CLI, developers can quickly generate components, services, modules, and other Angular artifacts, ensuring a consistent project structure and development workflow. It also offers built-in support for best practices like TypeScript, Webpack, and Angular Material.

Node.js is a crucial part of the ChatApp-Angular project, serving as the runtime environment for the server-side logic. It allows for the creation of a lightweight and efficient backend using JavaScript. In this chat application, Node.js, along with tools like `json-server`, is used to manage and serve the application's data, handle API requests, and ensure real-time data updates. This setup facilitates seamless communication between the client (Angular) and the server, ensuring a responsive and interactive user experience.

The app is designed to be user-friendly and accessible, with a clean and modern interface that adapts seamlessly to different devices and screen sizes. Whether users are chatting on their smartphones, tablets, or computers, the chat App ensures a consistent and enjoyable experience.

FUNCTIONALITY

- User Registration and Login: Allow users to create accounts and log in securely.

- Chat and login user Database: Maintain a comprehensive database of chat along with login details of user and their password in json server database.

- User Details: Display detailed information about each user, including their name , email id, phone number, etc

- User Profile: Allow users to view and edit their profile, manage their details, and view their chats with other users.

- Community Features: Include forums or chat rooms where users can whatever they want and interact with each other.

- Accessibility: Ensure the app is accessible to users with disabilities by following accessibility guidelines.

- Responsive Design: Design the app to be responsive and compatible with various devices and screen sizes.

HARDWARE / SOFTWARE REQUIREMENTS

Hardware Requirements

- Processor: Minimum dual-core CPU

- Memory: At least 4 GB of RAM

- Storage: 100 MB of available disk space

Software Requirements

- Operating System: Windows, macOS, or Linux

- Node.js: Version 16.0.0 or higher

- React : Version 16.0.0

- npm: Node Package Manager, included with Node.js installation

- Code Editor: Visual Studio Code or any preferred IDE

PROJECT OBJECTIVE

The primary objective of the ChatApplication project is to create a real-time, user-friendly chat application utilizing Angular for the frontend and Node.js for the backend. This project aims to demonstrate how to build a dynamic and interactive web application that facilitates seamless communication between users.

Key features include user authentication, message exchange, and real-time updates, showcasing the capabilities of modern web development frameworks and tools. The project also serves as a practical example for learning and understanding the integration of Angular and Node.js in building robust applications.

The objective of the ChatApp-Angular project is to develop a functional and interactive chat application using Angular for the frontend. The application aims to provide real-time messaging capabilities, user authentication, and a user-friendly interface. It leverages Angular's powerful features for building dynamic and responsive web applications, alongside Node.js and `json-server` for backend services and data management.

PROJECT FLOW

Project Flow of ChatApp-Angular

1. Setup and Installation:

   - Clone the repository or download the ZIP file.

   - Open the project in Visual Studio Code.

   - Run `npm install` to install dependencies.

2. Running the Server:

   - Use `npx json-server --watch db.json` to start the backend server, which handles user authentication and message data.

3. Development Server:

   - Start the Angular development server with `ng serve`.

   - Navigate to `http://localhost:4200/` to access the application.

4. Code Scaffolding:

   - Use Angular CLI commands like `ng generate component component-name` to create new components and other Angular artifacts.

5. Building and Testing:

   - Build the project using `ng build`.

   - Run unit tests with `ng test` and end-to-end tests with `ng e2e`.

PROJECT OUTCOME

Project Outcome of ChatApplication

The ChatApplication project aims to deliver a fully functional real-time chat application with the following outcomes:

1. User Authentication: Secure login and registration functionality using credentials stored in `db.json`.

2. Real-Time Messaging: Real-time chat capabilities allowing users to send and receive messages instantly.

3. Responsive UI: A user-friendly and responsive interface built with Angular.

4. Scalable Architecture: A modular and maintainable codebase leveraging Angular CLI for efficient development and deployment.

5. Testing and Build: Comprehensive unit and end-to-end testing setup, and a streamlined build process.

PROJECT TIME DURATION

The MovieX App is estimated to be completed within a timeframe of approximately three to four months. This timeline includes the planning, development, testing, and deployment phases of the project.

1. Planning Phase (1-2 weeks): This phase involves defining the project scope, objectives, and requirements, as well as creating a project plan and timeline.

2. Design Phase (2-4 weeks): During this phase, the app's architecture, user interface, and database design are created. This includes wireframing, prototyping, and finalizing the app's design.

3. Development Phase (3-5 weeks): In this phase, the actual development of the app takes place. Features such as user registration, searching of friends , can check on calls and contacts, and community interaction are implemented.

4. Testing Phase (1-2 weeks): The app is thoroughly tested for functionality, usability, and performance to ensure a bug-free and user-friendly experience.

5. Deployment Phase (0-1 weeks): Once testing is complete, the app is deployed to the production environment and made available to users.

6. Post-Launch Support and Maintenance: After the app is launched, there will be ongoing support and maintenance to address any issues, add new features, and ensure the app remains up-to-date with the latest technologies and movie databases.

Overall, the Chat App is expected to take approximately six to eight months from planning to deployment, with additional time for post-launch support and maintenance.

REFERENCES

- https://www.w3schools.com/php/
- https://www.javatpoint.com/html-tutorial
- https://www.canva.com/graphs/er-diagrams/
- https://ijcrt.org/papers/IJCRT22A6549.pdf
- https://github.com/
- https://nodejs.org/en
- https://getstream.io/chat/angular/tutorial/
- https://www.pubnub.com/blog/building-chat-application-using-angular/