

**A Project Report Submitted in Partial
Fulfilment of The Requirements for the
Degree of Master Of Computer
Application**

on

Snappy Chat Application

(Online chat application)

**Of
PROJECT
KCA-451
MCA-II year/IV Sem
Submitted by**

Sourabh Mishra	2201640140118
Nikhil Yadav	2201640140144

**Under the Supervision of
Ms. Amna Alam**



To the



Dr. A.P.J. Abdul Kalam Technical University, Lucknow Session 2023-24

INDEX

S.No	TITLE	PAGRE No
1.	INTRODUCTION	03
2.		04
3.	PROJECT SCOPE	05
4.	AIMS AND OBJECTIVE	06
5.	BRIEF MODULES DESCRIPTION	07
6.	TOOLS AND TECHNOLOGY USED	08 – 09
7.	METHODOLOGY	10 - 12
8.	I. Frontend	10
	II. Backend	10
	III. DFD	11
	IV. ERD	12
9.	EXPECTED TIME SCHEDULE (Gantt Chart)	13
10.	IMPACT OF PROPOSED SYSTEM IN ACADEMICS AND INDUSTRY	14
11.	ROLES AND RESPONSIBILITY	15
12.	PROS AND CONS	16
13.	REFERENCES	17
14.	CONCLUSION	18

INTRODUCTION

An online chat application is a digital platform that has revolutionized the way we communicate and connect with others in the digital age. In an era characterized by rapid technological advancements and an increasingly interconnected world, online chat applications have become an integral part of our daily lives. These applications serve as virtual meeting places where individuals from different corners of the globe can exchange messages, share media, and engage in real-time conversations with ease. Today Developers around the world are making efforts to enhance user experience of using Today Engineers around the world are making efforts to improve the user experience of the application and to improve the workflow of the developer to design applications to deliver projects and applications for the release of releases under a strict timeline.

GAP IN STUDY

➤ **Notifications:**

Implement notifications to alert users of new messages, mentions, or other relevant activities even when they are not actively using the application.

➤ **Typing indicators:**

Display typing indicators to show when other users are typing a message, enhancing the real-time nature of the chat.

➤ **User presence status:**

Show users' online/offline status and indicate when users are active or idle.

➤ **Integration with other services:**

Integrate the chat application with other services like third-party authentication providers, analytics platforms, or chatbot frameworks.

➤ **Moderation features:**

Implement moderation features such as message flagging, blocking users, or reporting inappropriate content.

➤ **Customization options:**

Provide users with customization options for themes, color, schemes and chat settings to tailor the application to their preferences.

PROJECT SCOPE

- Using a private network chat system or organizations.
- Ensuring the security of message and confidential data to be shared over the network.
- Keeping data confidential in a secure way.
- Creating a two-way communication system.
- Allow both group chat and private chat.
- To allow for easier and faster communication between people.
- Ensure unlimited data transfer without any size limit.
- Making people connect with others anytime, anywhere.

AIMS AND OBJECTIVE

- Real-time Communication

- User Authentication
- User Profiles.
- Message Persistence
- Group Chat Functionality
- Responsive Design
- Emoticons and Media Sharing
- Security Measures
- Search Functionality
- Read Receipts
- Callability and Performance
- Offline Messaging
- Privacy Settings
- User Interactivity
- Cross-platform Compatibility
- Analytics and Reporting
- Multilingual Support
- Integration with External Services
- Feedback Mechanism

Number of Modules

1) User Authentication and Authorization:

- Registration
- Password

➤ User Authentication

This module handles user registration, login, and security permissions to ensure that users can access the application securely.

2) Chat Module:

- One to One Module
- Group Module

If your application supports group chats, you'll need a module to create, manage, and participate in group conversations.

3) Set Avatar:

Update the UI to display the user's avatar on their profile or within the chat application after a successful upload

4) Security:

Ensure that the avatar upload process is secure by validating file types, setting file size limits, and implementing proper user authentication.

Tools and Technology Used

➤ **Software:**

- i. Visual Studio Code
- ii. Git

iii. Windows 10

➤ **Technologies:**

Frontend:

React JavaScript:

React.js is a popular open-source JavaScript library used for building user interfaces (UIs) and single-page applications (SPAs). React creates a virtual representation of the DOM in memory. It then calculates the most efficient way to update the real DOM based on changes in the virtual DOM, resulting in improved performance.

Backend:

Express JavaScript:

Express allows you to define routes to handle HTTP requests. You can create route handlers for various HTTP methods (e.g., GET, POST, PUT, DELETE) and specify URL patterns to match. This makes it easy to handle different types of requests and define the behaviour for each route.

Node JavaScript:

Node.js is an open-source, cross-platform runtime environment for executing JavaScript code outside the web browser. It allows developers

to use JavaScript for server-side scripting and building scalable network applications.

Mongo Database:

MongoDB stores data in documents, which are similar to JSON objects. Each document can have a flexible schema, meaning that different documents in the same collection can have different fields.

METHODOLOGY

➤ Frontend:

React JavaScript

React.js is a popular open-source JavaScript library used for building user interfaces (UIs) and single-page applications (SPAs). React creates a virtual representation of the DOM in memory. It then calculates the most efficient way to update the real DOM based on changes in the virtual DOM, resulting in improved performance.

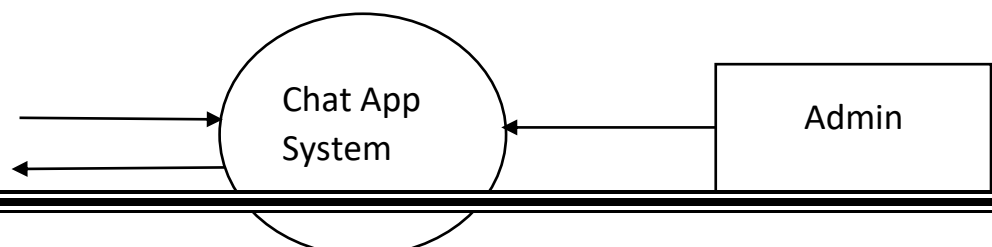
➤ **Backend:**

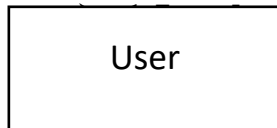
Express JavaScript

Express allows you to define routes to handle HTTP requests. You can create route handlers for various HTTP methods (e.g., GET, POST, PUT, DELETE) and specify URL patterns to match. This makes it easy to handle different types of requests and define the behaviour for each route.

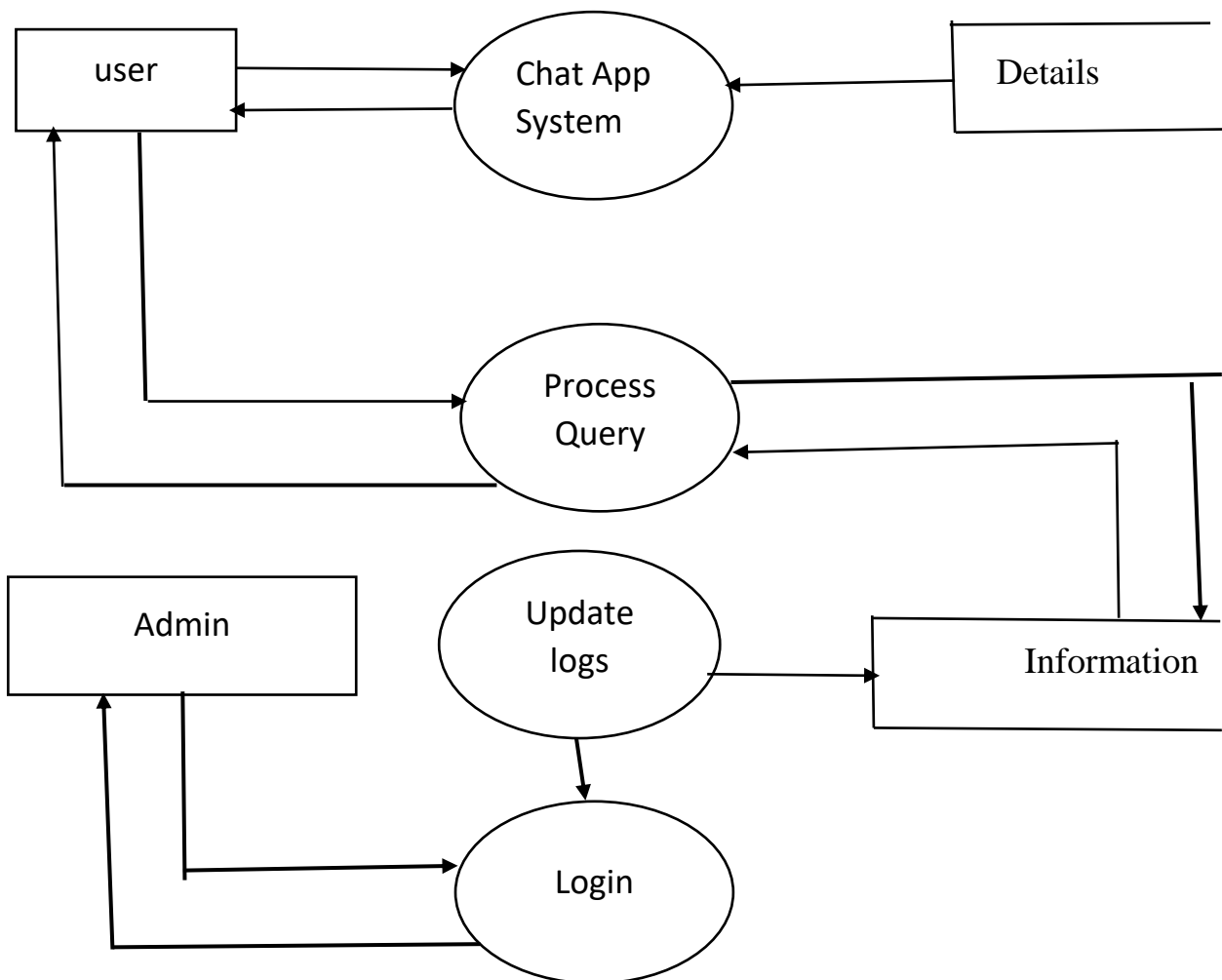
Data Flow Diagram

➤ **0-Level DFD**

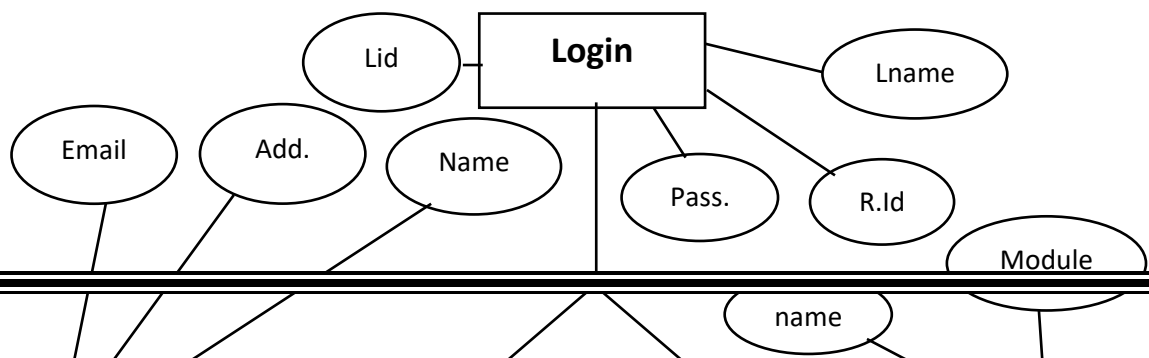




➤ **1-Level DFD**



ER-Diagram



EXPECTED TIME SCHEDULE (Gantt Chart)

Start Date		Snappy Chat Application		06-Oct To 04-jun (2024)									
S.No	Activity	Plan Start	Plan Eed	Fr	Sa	Su	Mo	Tu	We	Th			
		Actual Date	Actual End										
1	Project Planning	06-Mar-24	19-Mar-24	6	7	8	9	10	11	12			
				13	14	15	16	17	18	19			
				20	21	22	23						
2	Gather Requirment	20-Mar-24	26-Mar-24					24	25	26			
3	Designing Project	27-Mar-24	09-Apr-24	27	28	29	30	31					
				3	4	5	6	7	8	9			
				10	11	15	13	14	15	16			
4	Implementation	10-Apr-24	23-Apr-24	17		19		21	22	23			
5	Testing	24-Apr-24	07-May-24	24	25	26		28	29	30			
				1	2		4	5	6	7			
6	Deployment	08-Apr-24	24-May-24	8	9	10	11						
				15	16		18	19	20	21			
7	Maintenance	22-Apr-24	04-Jun-24	22	23			26		28			
				29	30				3	4			

IMPACT OF PROPOSED SYSTEM IN ACADEMICS AND INDUSTRY

➤ Impact on Academics

- Collaborative Learning
- Accessibility and Flexibility
- Resource Sharing
- Teacher-Student Engagement
- Event Announcements
- Project Collaboration
- Remote Work Support

➤ Impact on Industry

- Event Announcements
- Project Collaboration
- Remote Work Support
- Knowledge Sharing
- Client Communication
- Workflow Optimization

➤ Common Impacts

- Real-Time Interaction
- Centralized Information Hub
- Data Logging and Analysis
- Enhanced Connectivity
- Improved Documentation

ROLES AND RESPONSIBILITY

➤ **Project Manager**

- Overall project planning and coordination.
- Define project goals, scope, and timelines.
- Manage resources and budget.
- Coordinate communication between team members.

➤ **Backend Developer (Node.js and Express.js)**

- Develop server-side logic and APIs using Node.js and Express.js.
- Integrate with MongoDB for data storage.
- Ensure the security and scalability of the server-side components.
- Implement authentication and authorization mechanisms.

➤ **Frontend Developer (React.js)**

- Develop the user interface using React.js.
- Implement features for real-time communication.
- Integrate with backend APIs.
- Ensure a responsive and user-friendly interface.

➤ **Database Developer (MongoDB)**

- Design and implement the database schema in MongoDB.
- Optimize database queries for performance.
- Ensure data integrity and security.
- Handle data migration and backups.

➤ **Security Specialist**

- Conduct security assessments and audits.
- Implement security best practices in code and infrastructure.

PROS AND CONS

➤ **Pros:**

- Real-Time Communication
- JavaScript Stack
- Scalability
- Rich User Interfaces
- Rich User Interfaces
- NoSQL Database (MongoDB)
- Community and Documentation
- Cross-Platform Development

➤ **Cons:**

- Learning Curve
- SEO Challenges
- Full Stack Knowledge Required
- Complexity in State Management
- Scalability Concerns with MongoDB
- Security Considerations
- Potential Overhead for Small Projects
- Reliance on JavaScript

REFERENCES

➤ **Documentation**

- MongoDB Documentation
- Express.js Documentation
- React.js Documentation
- Node.js Documentation

➤ **Tutorials and Guides**

- Building a Real-Time Chat App with React, Node.js, and Socket.IO
- Build a Real Time Chat App with Node.js, Socket.io, and MongoDB

➤ **Video Tutorials:**

- Build a Chat App with React, Socket.io, and Node.js
- MERN Stack Tutorial (MongoDB, Express, React, Node.js)

➤ **GitHub Repositories**

- MERN Chat Application
- React + Node.js Chat Application

➤ **Online Courses**

- Full Stack Open - University of Helsinki
- MERN Stack Front To Back – Udemy

➤ **Books:**

- Pro MERN Stack:
- Learning React

CONCLUSION

People can successfully order the food using the proposed system. There will be a lesser requirement of staff at the back counter. The system will help in reduction of labour cost involved and also reduces the space required to set up cafeterias in the restricted area. As it is an automated system it is less probable to make any mistakes. The customers can avoid the long queues at the counter, with a reasonable speed of execution and maximum throughput.

