

Social Media Web App

1. INTRODUCTION

1.1 Project Overview

SPHERE is a dynamic and engaging social media web application inspired by Instagram, designed to foster seamless interaction, content sharing, and real-time communication among users. Built using the MERN stack, Sphere enables users to post images, like, comment, follow other users, save posts, and chat in real-time. The app emphasizes simplicity, responsiveness, and a modern user interface to enhance user engagement and satisfaction.

The goal of Sphere is to offer a full-featured, scalable platform that encourages authentic connections and meaningful content sharing while maintaining user privacy, security, and an intuitive experience across web devices.

1.2 Purpose

The purpose of the **Sphere App** is to provide users with a robust social media platform that allows seamless sharing of content and interaction with others through real-time messaging, likes, comments, and follows. Sphere aims to replicate and enhance the core features of modern social platforms while integrating efficient backend services, real-time updates, and strong security mechanisms to ensure a smooth and safe user experience.

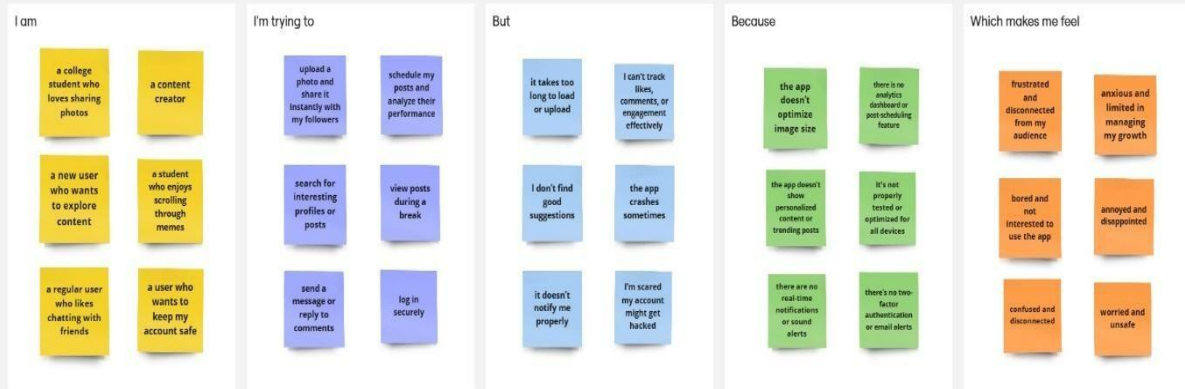
2. IDEATION PHASE

2.1 Problem Statement

Problem 1: How can we design and implement a robust social media web application using the MERN stack that enables real-time communication (messaging, notifications, story updates, etc.) across multiple users and devices, while maintaining high scalability, performance, and minimal latency?

Problem 2: How can we ensure the security, privacy, and data protection of users in a MERN stack-based social media application, while delivering an intuitive, seamless, and engaging user experience?

Customer Problem Statement Template



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a college student who loves sharing photos	upload a photo and share it	it takes too long to load or upload	the app doesn't optimize image size and lacks proper backend caching	frustrated and disconnected from my audience
PS-2	a content creator who relies on social media to grow my personal brand	schedule my posts and analyze their performance	I can't track likes, comments, or engagement effectively	there is no analytics dashboard or postscheduling feature in the app	anxious and limited in managing my growth
PS-3	a new user who wants to explore content	search for interesting profiles or posts	I don't find good suggestions	the app doesn't show personalized content or trending posts	bored and not interested to use the app


PS-4	a student who enjoys scrolling through memes	view posts during a break	the app crashes sometimes	it's not properly tested or optimized for all devices	annoyed and disappointed
PS-5	a regular user who likes chatting with friends	send a message or reply to comments	it doesn't notify me properly	there are no realtime notifications or sound alerts	confused and disconnected
PS-6	a user who wants to keep my account safe	log in securely	I'm scared my account might get hacked	there's no twofactor authentication or email alerts	worried and unsafe

2.2 Empathy Map Canvas Empathy Map

<p><u>Think & Feel</u></p> <ul style="list-style-type: none">• Concerned about privacy and how their data is used.• Want to stay updated with friends, trends,• Feel a sense of FOMO (Fear of Missing Out) if not active.• Think about how they are perceived by others.• Feel overwhelmed by too much content or negative news.	<p>Hear</p> <ul style="list-style-type: none">• Friends talking about viral content or trends.• Influencers promoting products or causes.• News outlets sharing breaking news via social media.• Feedback or comments from followers.• Peer pressure to join or stay active on certain platforms.
<p>See</p> <ul style="list-style-type: none">• Engaging visuals: reels, photos, stories, memes.• Ads and sponsored posts.• Notifications and alerts from the app.• Highlight reels of other people's lives.• Trending topics and hashtags.	<p>Say & Do</p> <ul style="list-style-type: none">• Share content: photos, videos, thoughts, opinions.• Comment on posts, like, react, share.• Recommend pages, tag friends, start conversations.• Express personal branding or identity.• Complain or praise features publicly
<p>Pain</p> <ul style="list-style-type: none">• Anxiety about likes, followers, or engagement.• Exposure to negativity, cyberbullying, or misinformation.• Time-wasting or feeling addicted.• Privacy concerns or data misuse.• Comparison leading to low self-esteem.	<p>Gain</p> <ul style="list-style-type: none">• Easy-to-use interface• Feeling connected and updated.• Gaining followers, recognition, or status.• Finding inspiration or entertainment.• Networking or business opportunities. • Sense of belonging to a community.

2.3 Brainstorming

Template



Brainstorm & idea prioritization

1

Problem statement

People want a simple and distraction-free platform to share photos, connect with friends, and engage with content. Existing social media apps are often cluttered and overwhelming. Our MERN stack Instagram clone solves this by offering a clean, user-friendly platform that focuses on core features like posting, liking, commenting, and following users.

2

Brainstorm

Komal

How to Give users a toggle to switch between light/dark themes?

Can we have Real-time Notifications?

Can we Fast uploads, even with low bandwidth?

Shreeya

Let users change font, bio style, or profile themes.

Achieve Two-Factor Authentication?

Ask Me Anything (AMA) Feature?

Shreya

Feature of Collaborative Posts (Two authors)

Live Streaming Possible ?

Suggested Friends Algorithm Based on mutuals and interests.

Vanshika

Search by location, hashtags, or photo content.

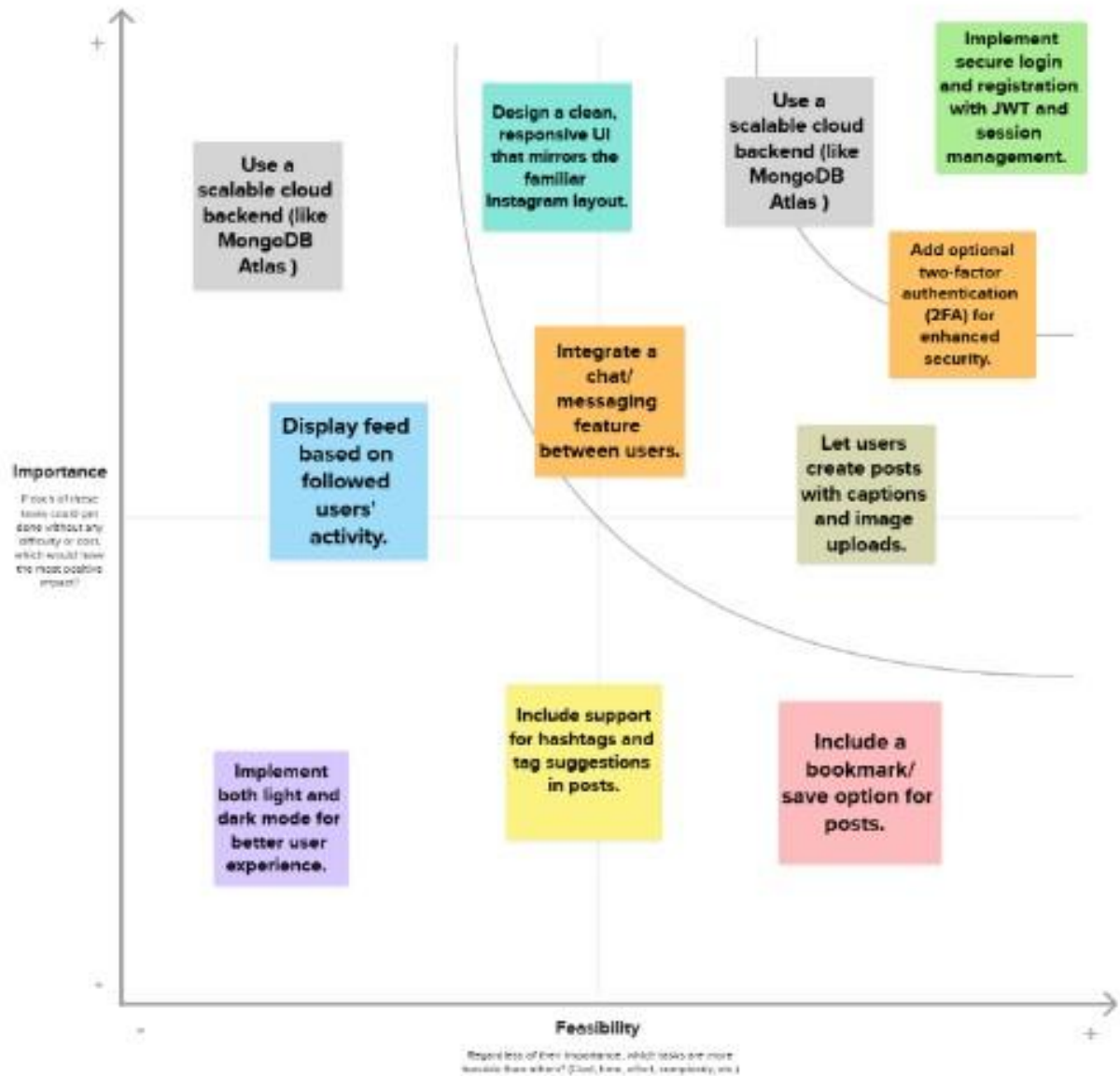
Private Account Option?

Verified Badge for Users

Group ideas



Prioritize



3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map



Customer Journey

Customer Journey Map

<u>Phase</u>	Awareness	Consideration	Decision	Service	Loyalty
<u>Customer Actions</u>	Sees ad or hears about app from a friend	Visits landing page, checks features & reviews	Downloads/ signs up for the app	Explores features, engages with content	Becomes a regular user, shares content, refers others
<u>Touchpoints</u>	Social media ads, influencer shoutouts, SEO/blogs	<u>Website</u> , app store page, <u>comparison</u> articles	App store, registration flow	In-app support, FAQ, onboarding guides	Push notifications, loyalty rewards, referral program
<u>Emotions</u>	Curious, intrigued	Interested, slightly <u>skeptical</u>	Excited, hopeful	Confused at first, then comfortable	Happy, confident, loyal
<u>Pain Points</u>	Doesn't know the brand, overwhelmed by options	Unsure about privacy, feature reliability	Signup friction, unclear UI	Bugs, delayed response to queries	Gets bored if not engaged, sees better alternatives
<u>Solutions</u>	Catchy branding, influencer marketing, SEO	Transparent policies, strong social proof, feature comparisons	Smooth onboarding, clear value proposition	Efficient support, engaging tutorials	Gamification, regular updates, user appreciation programs

3.2 Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Authentication	Login via form credentials Token/session-based authentication Password encryption

FR-4	User Profile Management	Create/Edit Photo Update Profile Manage Account
FR-5	Post Management	Create, edit, delete posts Support for text, image, video posts Save posts for later
FR-6	Story Management	Upload stories (images/videos) Auto-delete after 24 hours
FR-7	Real – Time Messaging	One-on-one and group chats Support for text, emoji, media Instant delivery with socket.io
FR-8	User Engagement Features	Like, comment, share posts Tag users Receive in-app notifications
FR-9	Search & Discovery	Search users, posts, tags Trending post section Follow/unfollow accounts
FR-10	Notifications	Like, comment, follow alerts Story viewed notifications Real-time updates

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Intuitive UI/UX with easy navigation, responsive design for all devices
NFR-2	Security	End-to-end encryption in chat, secure authentication, protection against data breaches
NFR-3	Reliability	Minimal downtime, robust error handling, consistent data sync

NFR-4	Performance	Fast response time (<200ms for user interactions), optimized database queries
NFR-5	Availability	Decent uptime guaranteed, ensured through cloudbased deployment and monitoring
NFR-6	Scalability	Support for growing number of users and data using horizontal scaling with MongoDB and Node.js cluster modules

3.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Social Media Web App DFDs:

Level-0 Description:

User:

The end-user who interacts with the social media web application to post content, like/comment, chat, and manage their profile. Post Content:

Handles user requests to create, edit, delete, and view posts. Updates and retrieves content from the database for display in user feeds. User Interaction:

Manages actions like likes, comments, follows, and unfollows. Updates the interaction records in the database.

Real-Time Chat:

Processes user messages and delivers them in real-time through socket communication. Manages chat history and active sessions. User Profile Management:

Handles user registration, login, logout, and profile updates. Manages personal user data securely.

Content Database:

Stores user-generated content such as posts, comments, and media.

User Database:

Stores user credentials, profiles, follower/following lists, and settings.

Media Storage Service:

External service (e.g., Cloudinary) used to store and retrieve images and videos uploaded by users.

Level-1 Description and Processes:

User:

The end-user who interacts with the social media web app by posting content, liking, commenting, following, and managing their profile.

Data Stores:

- User Data: Stores user credentials, profile info, settings, and followers/following relationships.
- Post Data: Stores user-generated content (posts, captions, media links, timestamps).
- Interaction Data: Stores likes, comments, shares, and user engagement records.
- Notification Data: Stores notification messages and their delivery status.

Processes (Level-1): 1.

User Interface (UI):

Handles all interactions between the user and the app. Displays feeds, posts, notifications, and profile information.

2. User Authentication (Auth):

- Processes login, signup, logout, and password recovery.
- Reads from and writes to User Data.

3. Content Manager:

- Handles creating, editing, deleting, and viewing posts. ○

Interacts with Post Data and Media Storage Service.

4. Interaction Handler:

- Manages likes, comments, and shares.
- Updates Interaction Data and can trigger notifications.

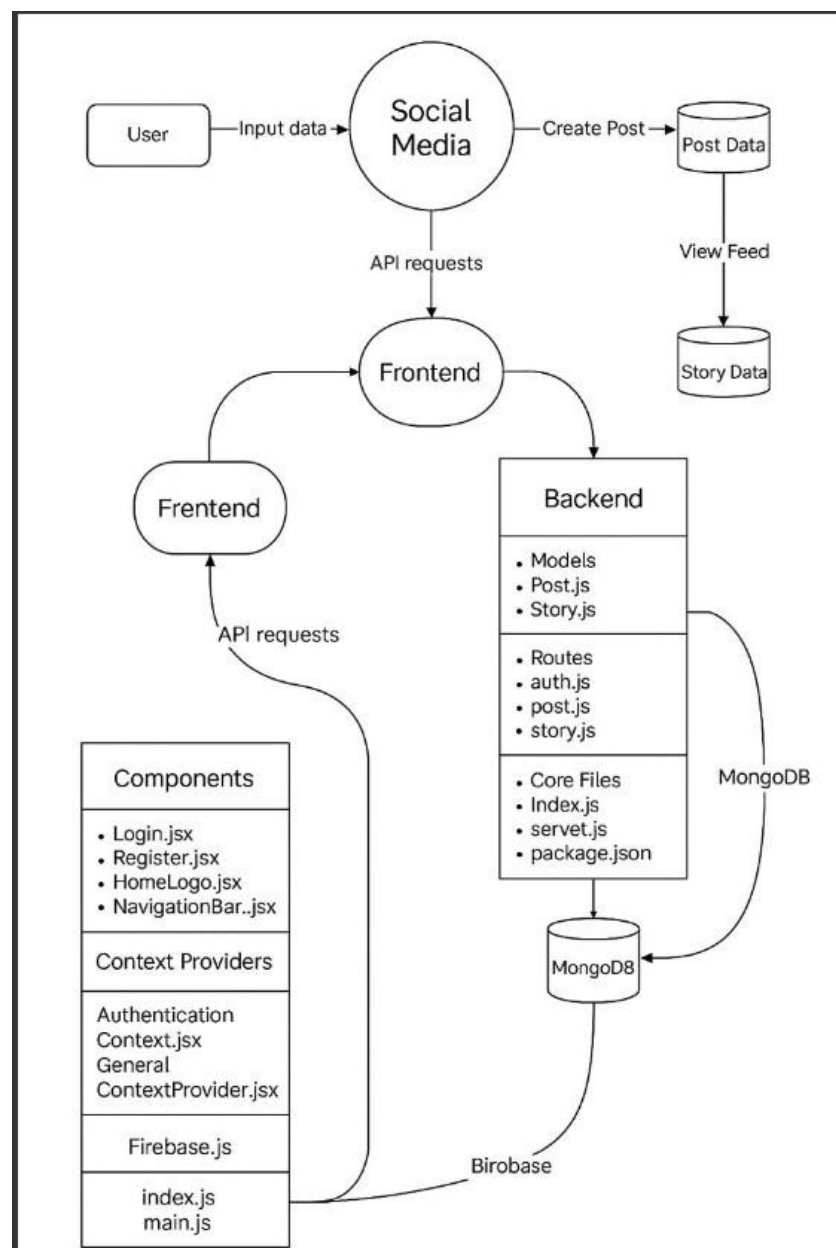
5. Profile Manager:

- Allows users to update profile details and manage follower relationships.
- Reads from and updates User Data.

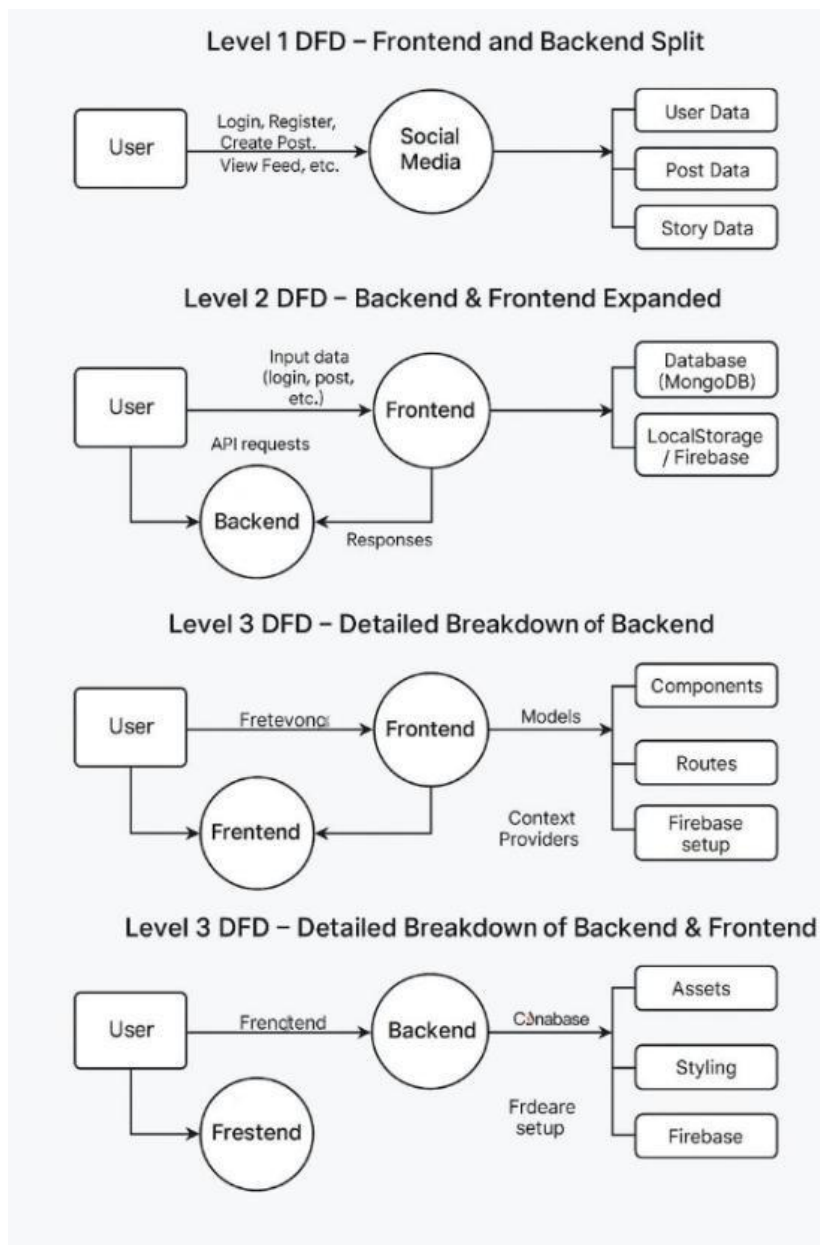
6. Feed Generator:

- Aggregates and displays posts for the user feed based on followers, interactions, and timestamps.
- Pulls data from Post Data and Interaction Data.

Level – 0 DFD



Level – 1 DFD



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Web user)	Account Management	USM-1	As a user, I want to create an account to join the platform	I can enter personal details, choose a username/password,	High	Sprint 1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
				and successfully register.		
Customer (Web user)	Account Management	USM-2	As a user, I want to log in to my existing account.	I can log in using my credentials and access my profile and feed.	High	Sprint 1
Customer (Web user)	Account Management	USM-3	As a user, I want to reset my password if I forget it.	I can request a password reset link and reset it through email.	Medium	Sprint 2
Customer (Web user)	Content Creation	USM-4	As a user, I want to create and share posts with media and captions.	I can upload media, write a caption, and post it successfully.	High	Sprint 1
Customer (Web user)	Content Interaction	USM-5	As a user, I want to like and comment on other users' posts.	I can click a like button and add comments that are visible under posts.	High	Sprint 1
Customer (Web user)	Profile Management	USM-6	As a user, I want to view and edit my profile.	I can update profile picture, bio, and personal information.	Medium	Sprint 2
Customer (Web user)	Search & Discover	USM-7	As a user, I want to search for users and follow them.	I can search users by name/username and follow/unfollow them	High	Sprint 3
Customer (Web user)	Feed/Timeline	USM-8	As a user, I want to view a feed of posts from people I follow.	I see a chronological/tailored list of posts from followed users.	High	Sprint 3

3.4 Technology Stack

Table- : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, React.js
2.	Application Logic-1	Registration, Login, Profile Setup	Node.js, Express.js
3.	Application Logic-2	Real-Time Messaging, Notifications	Socket.io, Node.js
4.	Application Logic-3	Newsfeed, Posts, Likes, Comments	Node.js, Express.js
5.	Database	User Info, Posts, Comments	MongoDB
6.	Cloud Database	Hosted Cloud DB	Firebase Firestore / MongoDB Atlas
7.	File Storage	Store images, videos, profile picture	AWS S3 / Firebase Storage
8.	External API-1	Social Login APIs	Google API, LinkedIn API
9.	Infrastructure (Server / Cloud)	Cloud Deployment & Scaling	Local, Kubernetes, etc.

Table: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryptions, HTTPS etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Docker, Kubernetes

4.	Availability	Justify the availability of application (e.g. use of load	AWS Load Balancer
S.No	Characteris tics	Description	Technology
		balancers, distributed servers etc.)	
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	IndexedDB

4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem-Solution Fit canvas		Purpose / Vision	Version:	
Define CS, fit into CL	1. CUSTOMER SEGMENT(S) CS <ul style="list-style-type: none"> • Teenagers and young adults • Content creators and influencers • Communities and interest groups • Small businesses promoting services 	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> <ul style="list-style-type: none"> • Need for a more personalized content feed • Hard to manage multiple social accounts in one place • Concerns over mental health and digital well-being 	5. AVAILABLE SOLUTIONS AS <small>PROS & CONS</small> <ul style="list-style-type: none"> • Traditional platforms (Instagram, Facebook, Twitter) • Emerging platforms (BeReal, Mastodon) • Third-party account managers and analytics tools 	
	2. PROBLEMS / PAINS + ITS FREQUENCY PR <ul style="list-style-type: none"> • Endless scrolling leads to stress and fatigue • Users feel their data is being misused or sold. • Harassment, cyberbullying, and negativity are prevalent. • Comparison, low self-esteem, addiction to validation • Feeds and interactions are not tailored to real interests. 	9. PROBLEM ROOT / CAUSE RC <ul style="list-style-type: none"> • Platforms prioritize engagement and ad revenue over well-being. • Data is collected without full user understanding or consent. • Gamified interfaces are built to keep users addicted. • Users are unaware of how posts are ranked or why they see certain content. • Content moderation is reactive, not proactive. 	7. BEHAVIOR + ITS INTENSITY BE <ul style="list-style-type: none"> • Scrolls frequently throughout the day • Shares content, likes, reacts, and reposts • Joins online trends or challenges • Switches apps based on vibe or content preference 	Explore AS, differentiate
Focus on PR, tap into BE, understand RC	3. TRIGGERS TO ACT TR <ul style="list-style-type: none"> • Increasing time spent on social platforms • Growing demand for authentic content over curated posts • Backlash against invasive ads and shadow algorithms 	10. YOUR SOLUTION SL <ul style="list-style-type: none"> • MERN-based web app with AI-personalized feed and privacy controls • Optional "mindful mode" limiting usage time • Role-based content filters (e.g., creator vs. consumer view) • End-to-end encryption for messaging and privacy-first data model 	8. CHANNELS of BEHAVIOR CH <p>ONLINE</p> <p>Tech blogs, YouTube tech influencers, app store reviews</p> <p>OFFLINE</p> <p>Peer recommendations, school/college discussions, events</p>	Focus on PR, tap into BE, understand RC
Identify strong TR & EM	4. EMOTIONS EM <small>BEFORE / AFTER</small> <p>Before : Overwhelmed, addicted, insecure, privacy concerns</p> <p>After: In control, safe, expressive, connected authentically</p>			Extract online & offline CH of BE

4.2 Proposed Solution

Proposed Solution:

S.No.	Parameter	Description
-------	-----------	-------------

1.	Problem Statement (Problem to be solved)	Current social media platforms often struggle with user privacy, content overload, and algorithm opacity. Users feel overwhelmed, unsafe, and disengaged due to lack of personalization, toxic content, and data misuse. There is a need for a platform that enables secure, personalized, and mindful social interactions.
2.	Idea / Solution description	Develop a modern social media web app using the MERN stack with features like AI-powered personalized feeds, end-to-end encrypted messaging, role-based content filtering, and digital well-being tools (e.g., mindful mode, screen time insights). The app will be modular, privacy-first, and designed for positive engagement.
3.	Novelty / Uniqueness	Unlike mainstream platforms, this app focuses on ethical design and user mental well-being. Key unique aspects include: <ul style="list-style-type: none"> • Transparent algorithm settings • Mindful interaction design (usage limits, moodbased content filters) • Secure messaging with optional ephemeral chats • Creator vs. Viewer role-based content display
4.	Social Impact / Customer Satisfaction	By promoting safe, intentional, and authentic online interaction, the platform addresses rising concerns around mental health, digital overload, and privacy. This builds trust and long-term user loyalty, especially among younger and socially conscious users. The platform encourages healthy usage while empowering users with real control over their feed and data.
5.	Business Model (Revenue Model)	Revenue will be generated through: <ul style="list-style-type: none"> • Freemium model – basic free access with premium customization and analytics • Creator monetization tools with commission • Ad-lite subscriptions • Brand partnerships and featured content • Optional API access for developers/influencer tools

6.	Scalability of the Solution	The application uses a microservices architecture hosted on scalable cloud infrastructure. As the user base grows, individual components (e.g., feed engine, messaging, analytics) can scale independently. WebSockets and NoSQL databases (e.g., MongoDB) support real-time engagement and high concurrency.
----	-----------------------------	---

4.3 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture of Project:

1) Client:

- User: Represents the end-user who interacts with the Social Media Web app.
- 2) Social Media Web App: The front-end application accessed by the user through their browser.
- 3) **Browser:** The web browser used by the client to access the web application.

4) WebServer:

- Node.js: The JavaScript runtime environment used for building the server-side of the application.
- Express.js: A web application framework for Node.js, used to handle HTTP requests and route them to the appropriate handlers.

4) AppServer:

- React.js: The JavaScript library used for building the user interface of the web app. React components are rendered in the browser, providing a dynamic and responsive user experience.

5) Database:

- MongoDB: The NoSQL database used to store user information, stock data, transaction history, and other relevant data for the web app.

6) ExternalServices:

- Social Media Data API: includes API endpoints for features like authentication, post management, and stories. No third-party authentication (e.g., OAuth) or external APIs (like cloud storage or email services) are explicitly listed in the repository.

Interaction Flow:

1) Client Interaction:

- The user interacts with the social media web app through their browser.

2) Request Flow:

- The browser sends an HTTP request to the WebServer (Node.js with Express.js) to access the web app.

3) User Actions:

- The user interacts with the React components in the app
- These interactions result in API calls from the React app to the WebServer. **4)**

Data Operations:

- The WebServer handles these API calls and performs CRUD (Create, Read, Update, Delete) operations on the MongoDB database.

5) External Services Interaction:

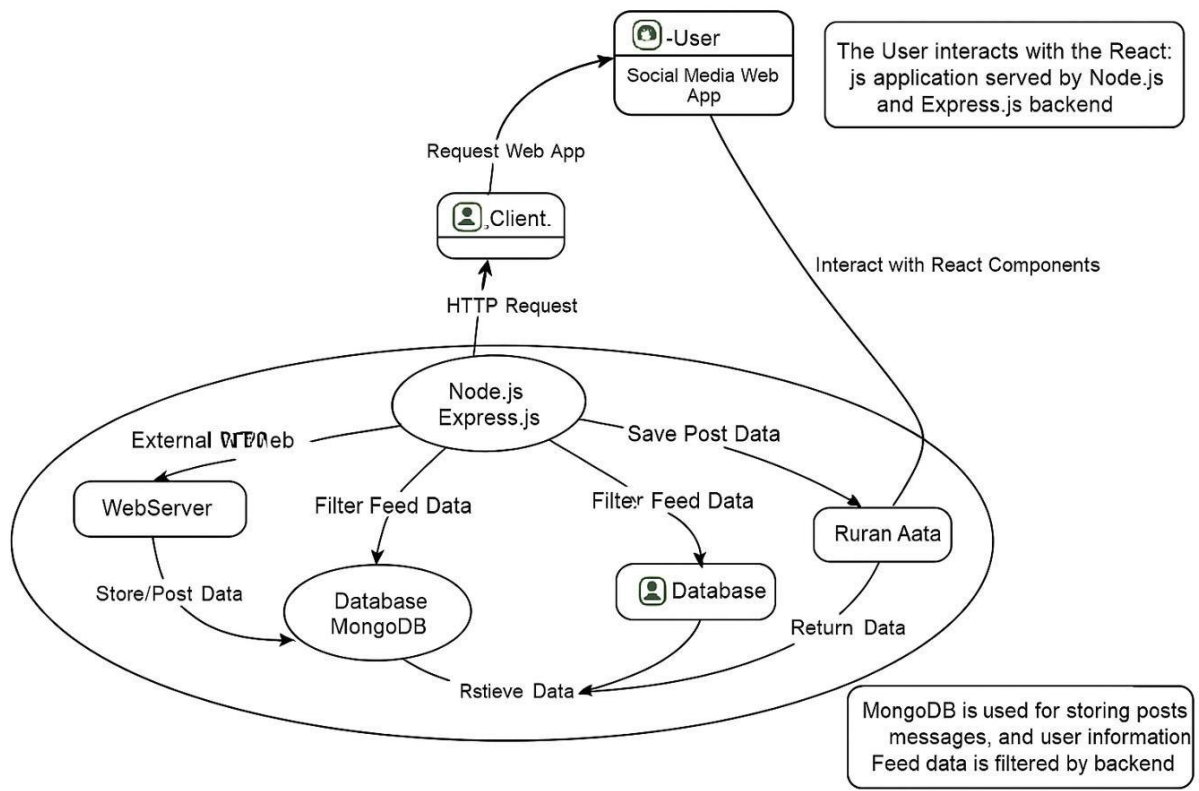
- It includes API endpoints for features like authentication, post management, and stories.
- The fetched data is saved in the MongoDB database for future use and quick access.

6) Response Flow:

The WebServer processes the data and returns the API responses to the React app.

- The React app updates the user interface based on the responses, providing the user with real-time data and feedback.

Solution Architecture Diagram:



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Authentication	USN-1	As a user, I can register and log in using my email and password.	10	High	Komal
Sprint-1	Profile Management	USN-2	As a user, I can create, edit, and view my profile details.	6	High	Shreya

Sprint-2	Post Management	USN-3	As a user, I can create, edit, delete, and view posts with captions and images.	8	High	Shreeya
Sprint-2	Like, Comment, Share	USN-4	As a user, I can like, comment on, and share posts with others.	8	Medium	Vanshika
Sprint-3	Explore and Follow Functionality	USN-5	As a user, I can explore posts from other users and follow/unfollow them.	12	High	Shreya
Sprint-4	Alerts	USN-6	As a user, I can chat in real-time using web sockets.	10	Medium	Komal
Sprint-4	Reporting and Analytics	USN-7	As a user, my account and messages should be secure and private.	6	High	Shreeya

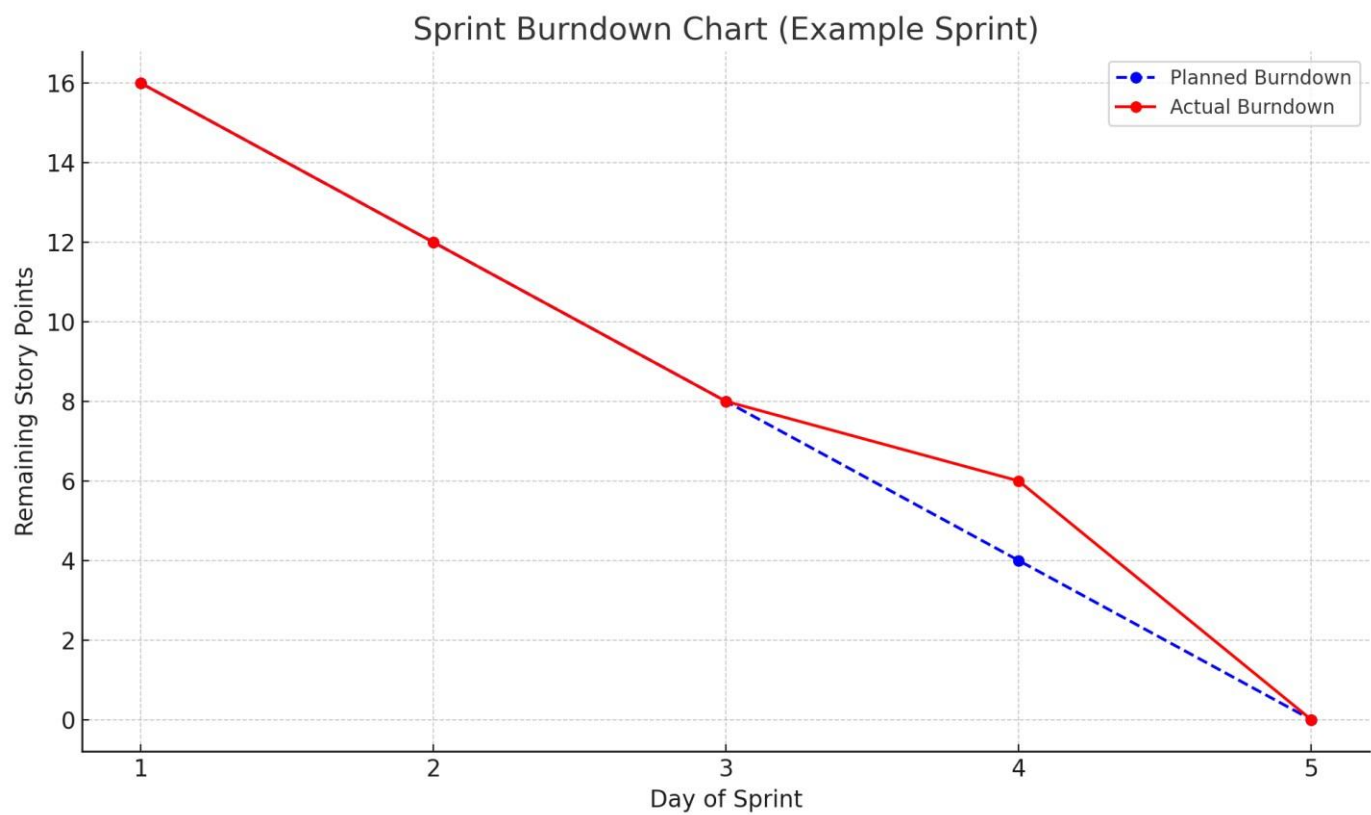
Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	16	5 Days	24 March 2025	29 March 2025	16	29 March 2025
Sprint-2	16	5 Days	30 March 2025	3 April 2025	16	3 April 2025
Sprint-3	12	5 Days	4 April 2025	9 April 2025	12	9 April 2025
Sprint-4	16	5 Days	9 April 2025	14 April 2025	16	14 April 2025

Sprint	Sprint Duration	Velocity	Average Velocity = Sprint duration/ velocity
Sprint 1	5 days	16	3.2 story points/ day
Sprint 2	5 days	16	3.2 story points/ day
Sprint 3	5 days	12	2.4 story points / day

Sprint 4	5 days	16	3.2 story points/day
----------	--------	----	----------------------

Burndown Chart:



6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Overview:

full-stack MERN (MongoDB, Express.js, React.js, Node.js) based social media web application designed for seamless user interaction and content sharing. The platform offers real-time feed updates, messaging, notifications, and user-friendly features for creating and managing posts, profiles, and connections.

Key Features:

- **User Authentication:** Register, log in, password reset.
- **Post Management:** Create, edit, delete, like, and comment on posts.
- **Real-Time Chat:** Direct messaging with online status.
- **Notifications:** Updates on likes, comments, new followers, etc.
- **User Profile:** Manage profile settings and view others’ profiles.
- **Search & Discover:** Search for users and trending content.

Technology Stack:

- **Frontend:** React.js, Redux, TailwindCSS
- **Backend:** Node.js, Express.js
- **Database:** MongoDB

Project Version: Version 1

Testing Period: 12/04/25 to 14/04/25 **Testing**

Scope:

List of Features and Functionalities to be Tested:

- User Registration & Login
- Post Creation, Editing, Deletion
- Like/Comment System
- Messaging & Chat
- Profile Management
- Search Functionality

List of User Stories or Requirements to be Tested:

- As a user, I want to register and log in securely.
- As a user, I want to create, edit, and delete posts.
- As a user, I want to interact with posts via likes and comments.
- As a user, I want to chat with other users in real time.
- As a user, I want to manage my profile settings and view others'.
- As a user, I want to search for content and users easily.

Testing Environment:

- URL/Location : http://localhost:6001
- Credentials : (EMAIL_API_KEY=saniyashreya04@gmail.com)

Test Cases:

<i>Test Case ID</i>	<i>Test Scenario</i>	<i>Test Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Pass/Fail</i>
TC-001	User Registration	1. Navigate to sign-up 2. Fill in user info 3. Submit form	User is successfully registered	Registered and redirected to profile page	Pass
TC-002	User Login	1. Open login page 2. Enter credentials 3. Click login	User logs in and is redirected to feed	User logged in successfully	Pass
TC-003	Create post	1. Log in 2. Open post form 3. Write and submit	Post appears in feed	Post created successfully	Pass
TC-004	Edit Post	1. Open existing post 2. Click edit 3. Save changes	Post updated	Changes reflected	Pass
TC005	Delete Post	1. Open post 2. Click delete	Post removed from feed	Post deleted	Pass

TC006	Like/Comment	1. Interact with a post	Likes/comments registered	Like/comment added	Pass
TC007	Chat Messaging	1. Open chat 2. Send message	Message delivered and displayed	Message seen by receiver	Pass
TC008	Profile Update	1. Open profile 2. Update info 3. Save	Profile info updated	Changes saved	Pass
TC009	Search Feature	1. Type in search 2. Select user or post	Results displayed	Relevant results shown	Pass

Bug Tracking:

Bug ID Steps to Severity Status Additional ID Description Reproduce Feedback

BG-001	Image not uploading	1. Create post 2. Attach image 3. Submit	High	Open	Error 500
BG-002	Chat not updating in real-time	1. Open chat 2. Send message	Medium	In Progress	Message only updates on refresh
BG-003	Profile picture not saving	1. Go to profile 2. Change pic 3. Save	Medium	Open	Reverts on reload

BG-004

Comment box not clearing

1. Add comment
2. Submit

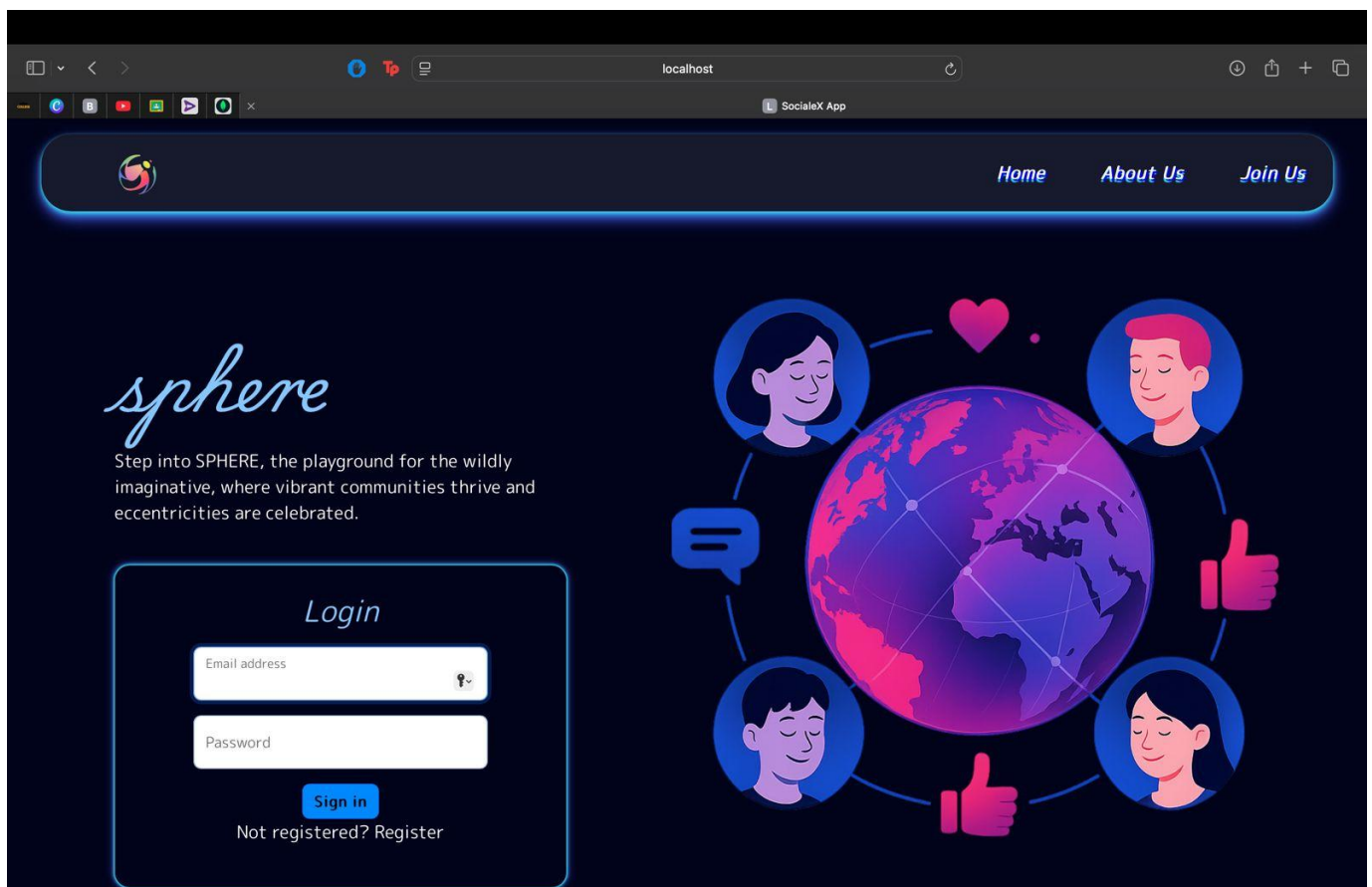
Low

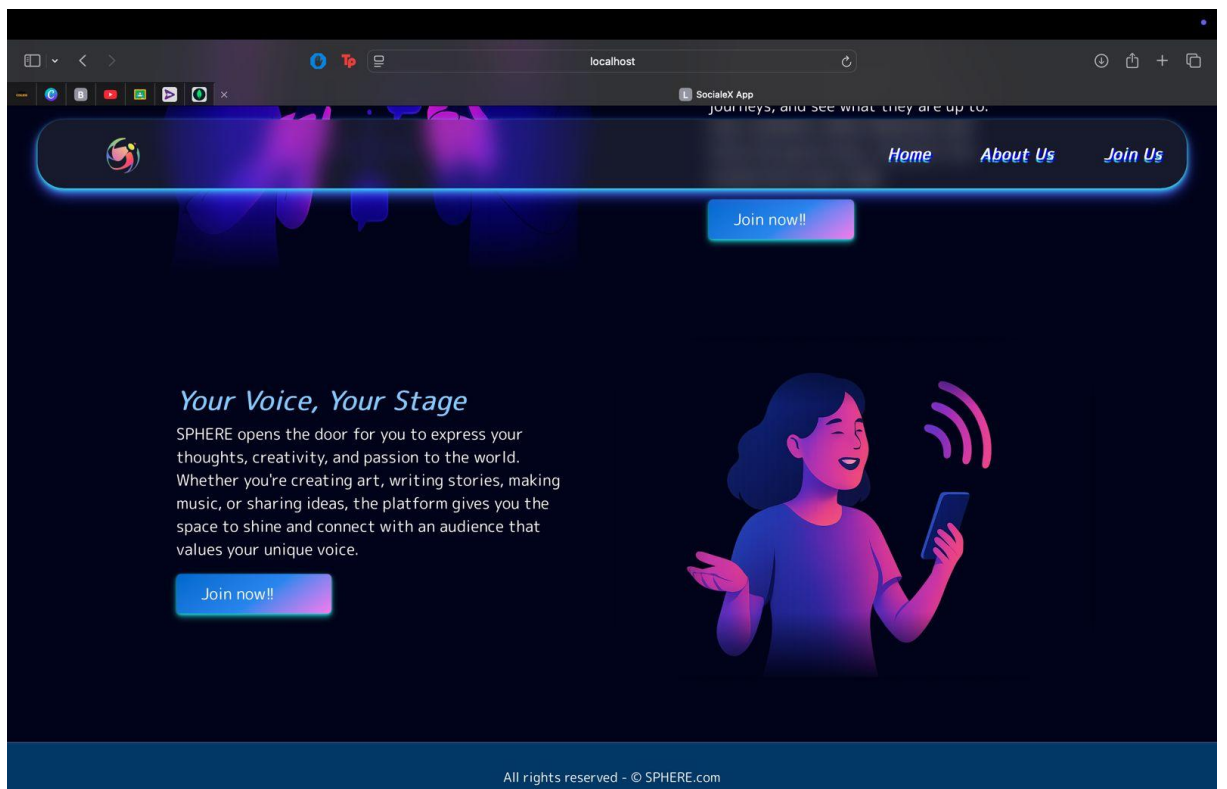
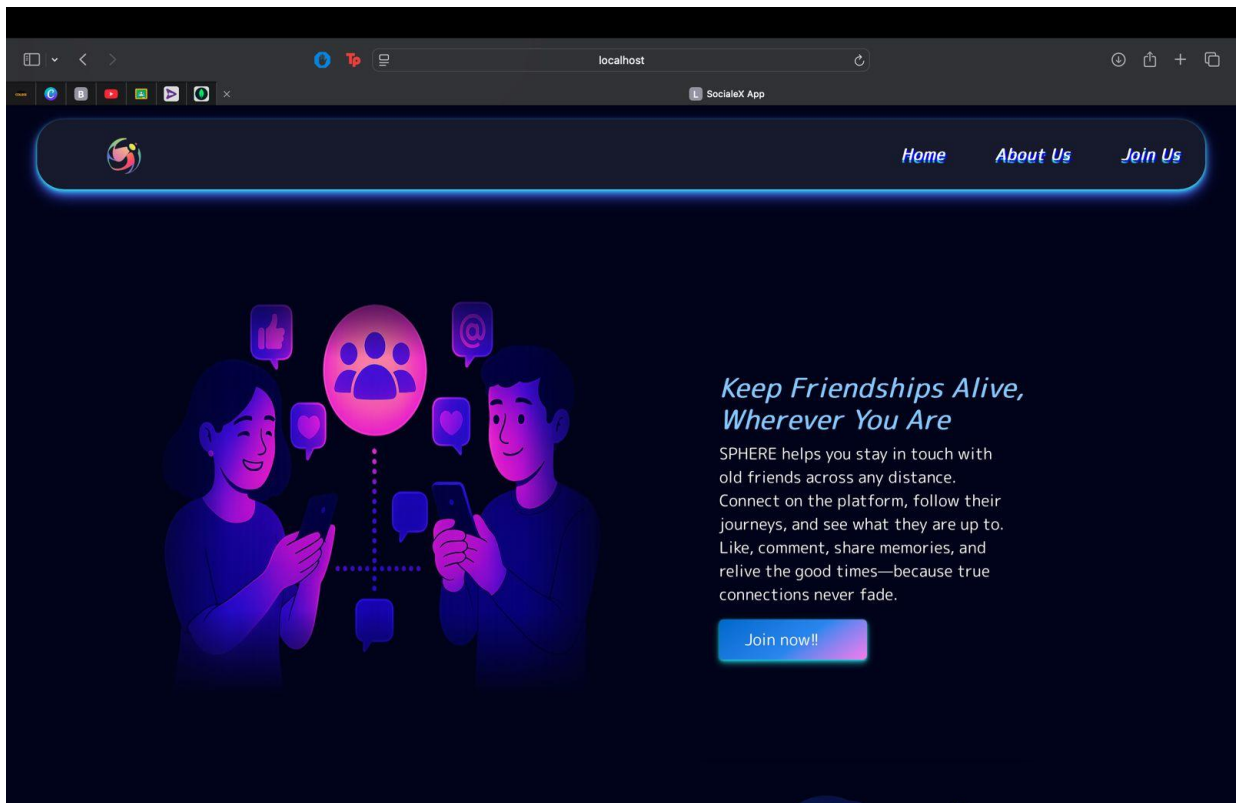
Closed

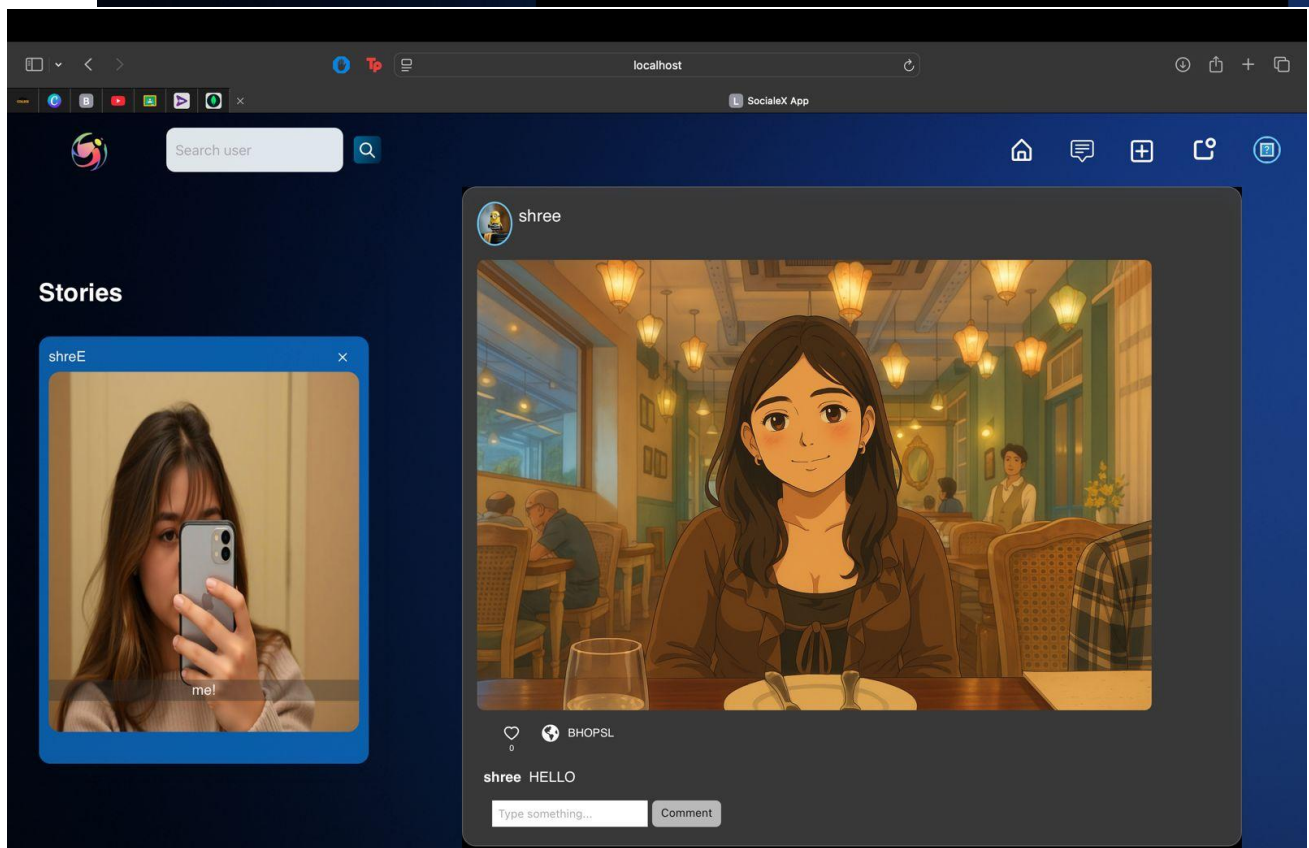
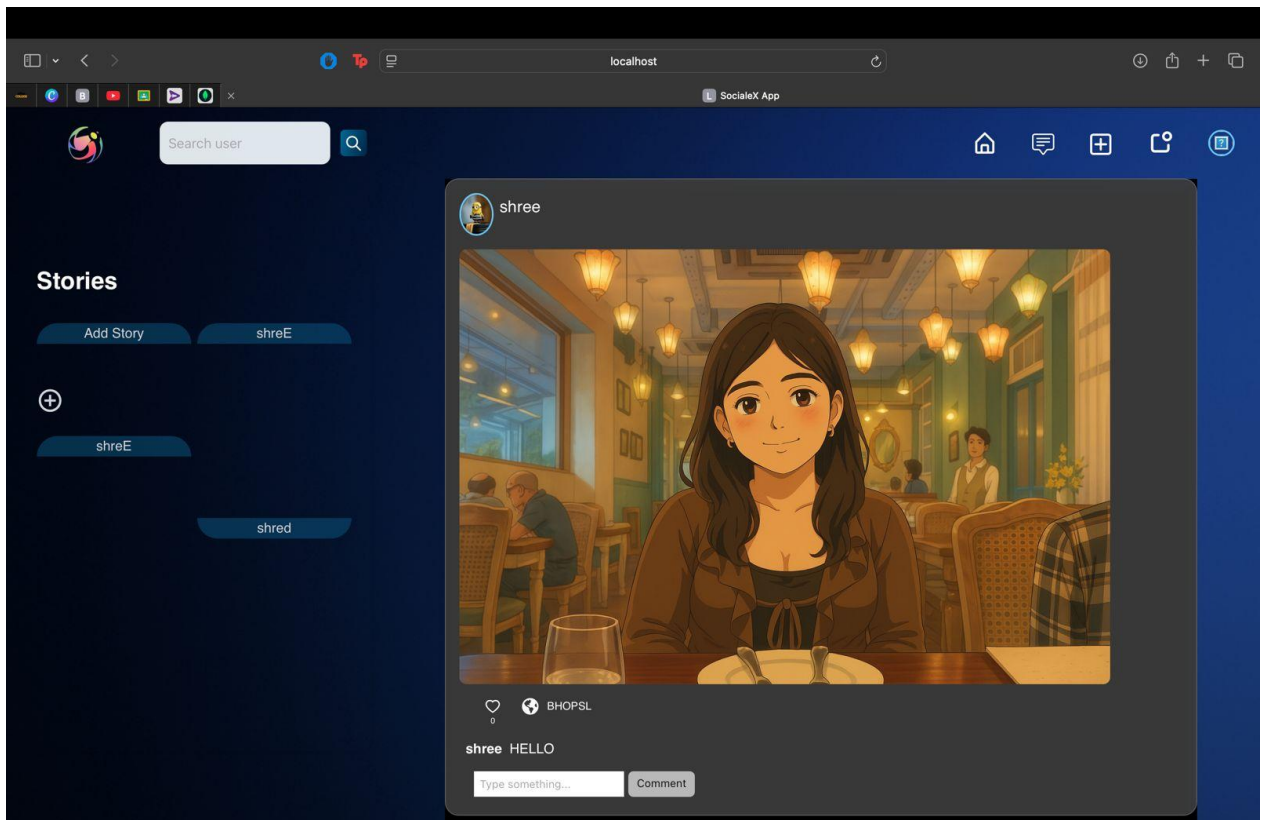
Comment posted but textbox stays filled

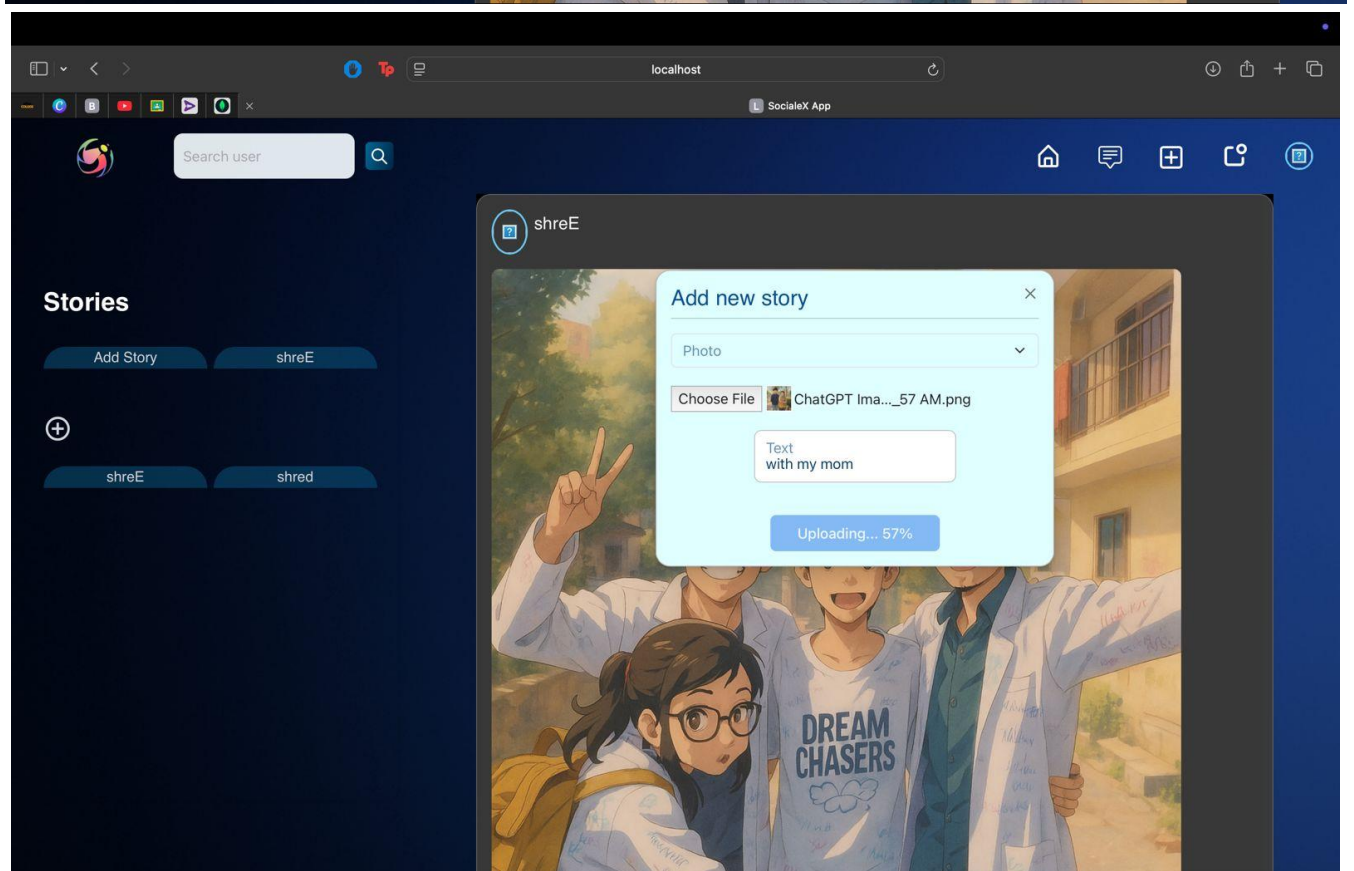
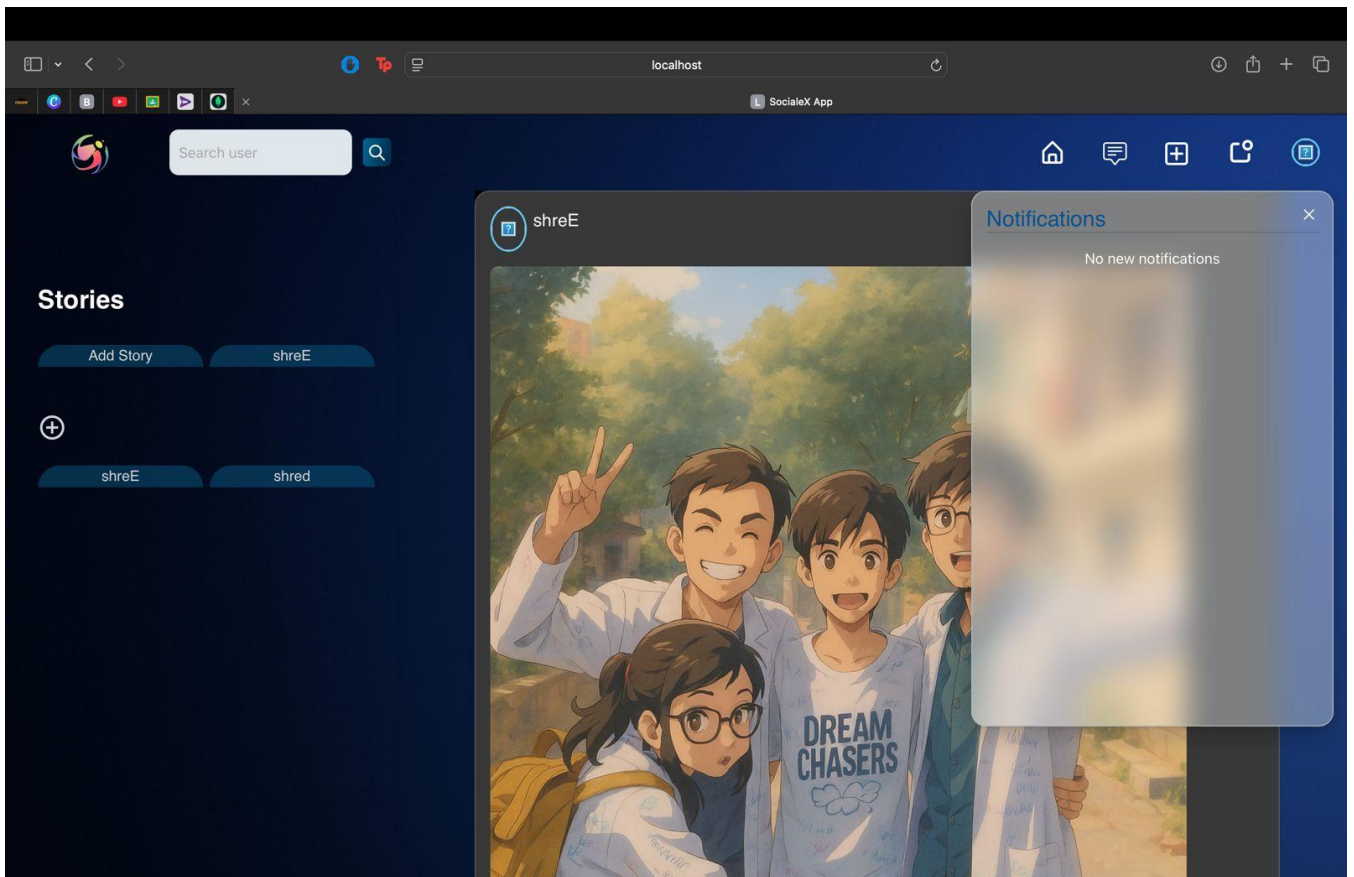
7. RESULTS

7.1 Output Screenshots









8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

- Real-time messaging with instant delivery and typing indicators
- Seamless content sharing: post images, like, comment, and share
- Responsive and intuitive user interface using React.js
- Secure user authentication using JWT and password encryption
- Efficient user interaction with post saving, following, and exploring features
- Cross-device accessibility — works on desktops, tablets, and mobile browsers
- Fast updates and scalable backend with Node.js and Express.js
- MongoDB provides flexibility for storing structured and unstructured data
- Personalized user feeds and suggested content
- Backend scalability supported by Socket.io for real-time communication

8.2 Disadvantages

- Occasionally, real-time messaging may experience minor delays under heavy traffic — optimization of socket connections is in progress
- Initial page load time may be high due to high-resolution images and media files — lazy loading and CDN integration is planned
- No mobile app support yet — currently web-only

9. CONCLUSION

The **Sphere App** delivers a modern, interactive social media experience through a wellintegrated MERN stack architecture. It provides all essential features of an Instagramlike platform, including real-time messaging, content interaction, secure user management, and a sleek UI. By combining functionality and scalability, Sphere offers a robust base for both social engagement and technical performance. It serves as a promising foundation for further enhancements and large-scale deployment

10. FUTURE SCOPE

- Development of native mobile applications for Android and iOS
- Introduction of story sharing and reels (short video) functionality

- Integration with cloud-based CDN for faster media delivery
- AI-powered feed personalization and content moderation
- Admin panel for content review and user management
- Integration with third-party services for advertisements and influencer insights
- Implementation of end-to-end encrypted messaging
- Dark mode and accessibility feature upgrade

11. APPENDIX

Documentation

- [MongoDB Documentation](#)
- [Express.js Documentation](#)
- [React Documentation](#)
- [Node.js Documentation](#)

Learning Resources

- [MERN Stack Tutorial](#)
- [JWT Authentication Tutorial](#)
- [React Router Documentation](#) ●
[Redux Toolkit Documentation](#)

API Resources

- [Socket.io Documentation](#)
- [Cloudinary API \(for image uploads\)](#)
- [Unsplash API \(optional image sourcing\)](#)