# Social Media Web App

## A PROJECT REPORT

**for**

**Major Project (KCA353)**

**Session (2023-24)**

Submitted by

Utsav Chakraborty

University Roll No 2200290140170

Submitted in partial fulfilment of the Requirements for the Degree of

# MASTER OF COMPUTER APPLICATION

**Under the Supervision of**

**Dr. Akash Rajak**

**Professor**



**Submitted to**

### Department Of Computer Applications

**KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206**

**(May 2024)**

# CERTIFICATE

Certified that **Utsav Chakraborty (2200290140170)** has carried out the project work having “**Social Media Web App**” (**Major Project KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date: Utsav Chakraborty**

**(2200290140170)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Akash Rajak Dr. Arun Kumar Tripathi

Professor Professor & Head

Department of Computer Applications Department of Computer Applications KIET Group of Institutions, Ghaziabad KIET Group of Institutions, Ghaziabad

# ABSTRACT

The Social Media Web Application, an Instagram clone, is designed to provide a platform for users to share photos, interact with posts, and follow other users. This application employs React JS for the frontend, ensuring a dynamic and responsive user interface, while Tailwind CSS is used for styling. Firebase is utilized for the backend, offering real-time database services and user authentication.

Key Features:

User Authentication: Secure login and registration for users, with Firebase handling authentication.

Post Creation and Management: Users can create, edit, and delete posts, including photo uploads.

Interactive Feed: Users can view, like, and comment on posts from the users they follow.

Responsive Design: The application is optimized for various devices, providing a seamless experience across desktops, tablets, and smartphones.

Real-time Updates: Firebase ensures that posts and interactions are updated in real-time across all user interfaces.

The Social Media Web Application aims to replicate key functionalities of Instagram, providing users with a familiar and engaging social media experience.

# ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr Akash Rajak** for his guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Utsav Chakraborty

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Certificate | | | i |
|  | Abstract |  | ii |
|  | Acknowledgement | | iii |
|  | Table of Contents | | iv-v |
|  | List of Tables | | vi |
|  | List of Figures | | vii |
| 1 | Introduction | | 8-9 |
|  | 1.1 | Background | 7 |
|  | 1.2 | Project overview | 7 |
|  | 1.3 | Objective | 9 |
|  | 1.4 | Key features | 10 |
|  | 1.5 | Scope of the project | 11 |
| 2 | Problem identification & feasibility study | | 12-14 |
|  | 2.1 | Problem Identification | 12 |
|  | 2.2 | Feasibility Study | 13 |
|  |  | 2.2.1 Technical Feasibility | 14 |
|  |  | 2.2.2 Operational Feasibility | 14 |
|  |  | 2.2.3 Economic Feasibility | 14 |
| 3 | Requirement Analysis | | 15-18 |
|  | 3.1 Introduction | | 11 |
|  | 3.2 Functional Requirements | | 11 |
|  | 3.3 Non-Functional Requirements | | 11 |
| 4 | Project Planning and Scheduling | | 19-21 |
|  | 4.1 Pert Chart | | 19-21 |
| 5 | Hardware and Software Specification | | 22 |
|  | 5.1 Hardware Specification | | 22 |
|  | 5.2 Software Specification | | 22 |
| 6 | Choice of Tools & Technology | | 23-30 |

|  |  |  |
| --- | --- | --- |
|  | 6.1 React JS | 23 |
|  | 6.2 Tailwind CSS | 23 |
|  | 6.3 Firebase | 23 |
|  | 6.4 Context Level Diagram | 24-30 |
| 7 | ER-Diagram | 31-32 |
|  | 7.1 Entity-relationship model | 31 |
|  | 7.2 Class Diagram | 32 |
| 8 | Database | 33 |
| 9 | Form Design | 34-37 |
| 10 | Testing | 25 |
|  | 10.1 Introduction | 25 |
|  | 10.2 Types of Testing | 25 |
|  | Future Scope and Further Enhancement of the Project 26 | |
|  | Conclusion & References | 27 |

**LIST OF TABLES**

|  |  |
| --- | --- |
| **Name of Table** | **Page** |
| Authentication | 34 |
| Create Post | 35 |
| Home Page | 36 |
| Registration Page | 37 |

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **Name of Figure** | **Page No.** |
| Pert Chart | 21 |
| 0 Level DFD | 27 |
| 1 Level DFD | 28 |
| 2nd Level DFD | 29 |
| Entity-relationship Model | 31 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Background**

The Social Media Web Application project aims to replicate the core functionalities of Instagram, providing a platform for users to share photos, follow others, and interact with posts. This application is designed to offer a user-friendly and engaging social media experience, addressing the increasing demand for social connectivity in the digital age. Social media platforms have revolutionized how people communicate, share moments, and stay updated with their social circles. With the proliferation of smartphones and the internet, users now seek seamless and real-time experiences, which this project aspires to deliver.

In recent years, the importance of social media in everyday life has grown exponentially. From personal connections to professional networking, social media platforms have become an integral part of our daily routines. By creating a platform that mimics the successful features of Instagram, this project intends to provide users with a familiar yet enhanced environment for social interaction. The focus on user experience and real-time interaction aims to make this platform a preferred choice for photo sharing and social engagement.

* 1. **Project Overview**

The project involves creating a web-based platform where users can create accounts, share photos, view the feed of the users they follow, and interact with posts. The frontend is powered by React JS for a dynamic and responsive user experience, while Tailwind CSS ensures a modern and consistent design. Firebase is used for the backend, providing real-time database services and secure user authentication. The combination of these technologies aims to offer a robust and scalable solution for social media interaction.

React JS, a popular JavaScript library for building user interfaces, is chosen for its component-based architecture and efficiency in managing the application state. Tailwind CSS is selected for its utility-first approach to styling, which allows for rapid development and a cohesive design language. Firebase, a comprehensive app development platform, provides essential backend services such as real-time database, authentication, and hosting. This technology stack ensures that the application is both performant and secure.

The user interface (UI) and user experience (UX) are critical components of this project. A well-designed UI ensures that users find the application visually appealing, while a well-thought-out UX guarantees ease of use and intuitive navigation. The application's design principles are centered around simplicity, responsiveness, and accessibility, ensuring that users across different devices and screen sizes have a consistent experience.

* 1. **Objective**

The primary objective of the Social Media Web Application include:

* User Engagement: Providing an interactive platform for photo sharing and social interaction. The application aims to keep users engaged through features like photo sharing, liking, commenting, and following other users. Engagement is a key metric for the success of social media platforms, and this project focuses on creating an engaging environment that encourages users to spend more time on the platform.
* Real-time Interaction: Ensuring updates and interactions are reflected in real-time. Real-time interaction is crucial for maintaining the dynamic nature of social media. By leveraging Firebase's real-time database capabilities, the application can provide instantaneous updates, ensuring that users see the latest content and interactions without delay.
* User Convenience: Offering a familiar and intuitive interface for social media users. A familiar interface reduces the learning curve for new users, making it easier for them to start using the platform. The interface design takes inspiration from popular social media applications, ensuring that users feel comfortable and confident navigating the app.
  1. **Key Features**

1. **User Authentication:**
2. Secure login and registration: Ensuring that user data is protected through secure authentication methods. Users can create accounts using their email and password, and the authentication process includes measures like email verification and password encryption to enhance security
3. Password encryption: All passwords are encrypted using industry-standard encryption techniques, ensuring that user credentials are not stored in plain text and are protected from unauthorized access.
4. **Post Creation and Management:**
5. Users can create, edit, and delete posts: Providing flexibility for users to manage their content. Users have complete control over their posts, with options to create new posts, edit existing ones, and delete those they no longer wish to keep.
6. Upload and display photos: Allowing users to share their photos with their followers. The photo upload feature supports various image formats and ensures that photos are displayed in high quality.
7. **Interactive Feed:**
8. View, like, and comment on posts: Enabling users to interact with content shared by others. The feed displays posts from users that one follows, and interactions like likes and comments are prominently featured to encourage engagement.
9. Real-time updates: Ensuring that interactions are updated instantaneously. Firebase's real-time database ensures that any new likes, comments, or posts are immediately visible to users without needing to refresh the page.
10. **Responsive Design:**
11. Ensure the application is visually appealing and functionally on various devices and screen sizes: The design is responsive, adapting to different screen sizes and orientations to provide a seamless user experience across desktops, tablets, and mobile devices.
12. **Real-time Update**s:
13. Firebase ensures posts and interactions update in real-time: Leveraging Firebase's capabilities, the application maintains real-time synchronization of data, providing users with up-to-date information and interactions.
    1. **Scope of the Project**

The scope of this project includes the development of a full-featured social media web application. This entails the implementation of the core functionalities such as user authentication, post creation and management, interactive feed, and real-time updates. The project's initial phase focuses on these essential features to ensure a solid foundation.

Future enhancements may include advanced features such as direct messaging, stories, and analytics for user engagement. Direct messaging would allow users to communicate privately, while stories provide a way to share ephemeral content. Analytics can offer insights into user behavior and engagement, helping to refine the platform and enhance user experience.

Additionally, the scope includes ensuring high performance and scalability to accommodate a growing user base. The application is designed with scalability in mind, utilizing Firebase's infrastructure to handle increased traffic and data load. Performance optimizations are implemented to ensure that the application remains responsive and fast, even as the number of users and interactions grows.

By addressing these aspects, the project aims to create a comprehensive social media platform that meets the needs of modern users, providing a reliable, engaging, and feature-rich environment for social interaction.

* 1. **Importance of the Project**

The Social Media Web Application project holds significant importance in today’s digital landscape. Social media platforms are pivotal in shaping communication, culture, and commerce. By developing a platform that encompasses the essential features of Instagram, this project not only meets the recreational needs of users but also supports business and marketing efforts. Social media's influence on consumer behavior and decision-making highlights the necessity of having a reliable and engaging platform.

Moreover, this project serves as a learning and innovation opportunity for developers and stakeholders. It allows for the exploration of modern web development technologies such as React JS, Tailwind CSS, and Firebase. This practical experience can be invaluable for personal and professional growth in the tech industry.

* 1. **Challenges and Considerations**

While developing the Social Media Web Application, several challenges and considerations must be addressed:

1. Scalability: Ensuring the platform can handle a large and growing user base without performance degradation.
2. Security: Implementing robust security measures to protect user data and prevent unauthorized access

.

1. User Experience: Creating an intuitive and engaging user interface that encourages interaction and retention.
2. Real-time Performance: Ensuring that updates and interactions occur in real-time without lag or delay.
3. Compliance: Adhering to legal and regulatory requirements related to data privacy and user content

**CHAPTER 2**

**PROBLEM IDENTIFICATION & FEASIBILITY STUDY**

### Problem Identification

In the rapidly evolving digital landscape, social media platforms play a crucial role in connecting people, sharing content, and fostering online communities. However, several common issues persist in existing social media platforms, impacting user experience and satisfaction. The Social Media Web Application project seeks to address these issues through innovative solutions, aiming to create a superior platform that enhances user engagement and security while simplifying the user interface.

1. Delayed Updates: Many social media platforms suffer from latency issues where user interactions and content updates are not reflected in real-time. This delay can disrupt the user experience, causing frustration and reducing engagement. Users expect instantaneous feedback, likes, comments, and new posts, which necessitates a robust real-time update mechanism.
2. Security Vulnerabilities: Security is a significant concern for users of social media platforms. Data breaches, unauthorized access, and the misuse of personal information have plagued many platforms. Ensuring robust security measures, such as encryption and secure authentication, is vital to protecting user data and maintaining trust.
3. Complex User Interfaces: A complex and cluttered user interface can detract from the user experience. Users prefer a clean, intuitive interface that is easy to navigate. Overly complicated interfaces can lead to user frustration, lower engagement, and ultimately, user attrition. Simplifying the user interface while maintaining comprehensive functionality is crucial.
4. User Engagement: Maintaining high levels of user engagement is a challenge for many platforms. Features that fail to capture users' interests or are too difficult to use can lead to reduced interaction and lower retention rates. It is essential to design features that are not only functional but also engaging and user-friendly.
5. Scalability Issues: As user bases grow, many platforms face difficulties scaling effectively. This can lead to slower performance, increased downtime, and a negative user experience. Ensuring that the platform can scale efficiently to accommodate growth is critical for long-term success.

By addressing these problems, the Social Media Web Application aims to create a platform that is responsive, secure, and user-friendly, offering an enhanced experience for all users.

### Feasibility Study

A feasibility study is essential to assess the viability and practicality of developing the Social Media Web Application. This study covers technical, economic, operational, and legal considerations to ensure the project is feasible and sustainable in the long term.

### Technical Feasibility

* **Technology Stack**:

1. Frontend: React JS is chosen for its efficiency, component-based architecture, and dynamic rendering capabilities. React JS allows for the creation of a highly responsive user interface that updates in real-time.
2. Styling: Tailwind CSS is selected for its utility-first approach to styling, which simplifies the design process and ensures a consistent and modern look across the application.
3. Backend: Firebase is used for its comprehensive suite of services, including real-time database, authentication, and hosting. Firebase's real-time capabilities are crucial for providing instantaneous updates and interactions.

* **API Integration:** Integration with Firebase services is critical for real-time updates and secure user authentication. This includes using Firebase Authentication for secure login and registration processes, and Firebase Realtime Database for storing and syncing user data in real-time.
* **Scalability:** The chosen architecture must be evaluated for scalability to handle potential increases in user base and content. This includes assessing the capacity of Firebase to manage large volumes of data and traffic without compromising performance. Efficient coding practices and optimization techniques are implemented to ensure that the application can scale seamlessly.
* Performance Optimization: Ensuring optimal performance involves minimizing load times, reducing latency, and optimizing the rendering process. This includes leveraging React JS's virtual DOM for efficient updates and Tailwind CSS for fast, responsive design.
* Reliability and Uptime: The platform must be designed for high availability and reliability. This involves setting up Firebase Hosting for robust hosting solutions, implementing automated backups, and ensuring failover mechanisms are in place to minimize downtime

### Operational Feasibility

* **Resource Availability:** The project requires skilled developers with expertise in React JS, Tailwind CSS, and Firebase. Ensuring the availability of these resources is critical for the successful completion of the project. This includes not only developers but also UI/UX designers, testers, and project managers.
* **Timeline:** A realistic development timeline must be established, considering potential challenges such as technical difficulties, integration issues, and testing phases. A phased development approach can be adopted, starting with the core functionalities and progressively adding advanced features.
* **Operational Impact:** The operational impact includes the positive effects on user engagement and satisfaction. A well-designed platform with real-time capabilities and a user-friendly interface can significantly enhance user interaction and retention. Regular feedback from users during development can help in fine-tuning the platform to better meet their needs.
* **Training and Support:** Adequate training and support mechanisms must be in place for both the development team and the end-users. This includes comprehensive documentation, user guides, and a support system to address any issues that arise post-deployment.

### 2.2.3 Economic Feasibility

* **Cost Estimation:** Estimate the costs associated with development, hosting.A cost-benefit analysis helps in understanding the financial viability of the project. This involves comparing the projected costs against the potential revenue and benefits, ensuring that the project is economically sustainable in the long run.
* **Return on Investment (ROI):** Calculating the expected ROI is important for stakeholders. This includes projecting the time frame within which the platform will start generating profit and the expected financial gains over a specific period.

**CHAPTER 3**

**REQUIREMENT ANALYSIS**

## Functional Requirements: Functional requirements define the specific behaviors and functionalities that the application must support. These are essential to ensure that the application meets its intended purpose and provides a satisfactory user experience.

**User Authentication:**

* + Secure login and registration

1. User Registration: Users must be able to create an account using an email address and a password. The registration process should include email verification to confirm the user's identity.
2. User Login: Users should log in using their registered email and password. Implementing a secure login process with measures such as account lockout after multiple failed attempts enhances security.
3. Password Recovery: Users must have the option to recover or reset their password through a secure method, such as receiving a reset link via email.
   * Password encryption:
4. Encryption Standards: Passwords must be encrypted using industry-standard encryption techniques such as bcrypt. This ensures that passwords are not stored in plain text and are protected from unauthorized access.

**Post Creation and Management:**

* + Create, Edit, and Delete Posts:

1. Post Creation: Users should be able to create posts, including adding text descriptions and uploading photos. The interface for creating posts should be intuitive and straightforward.
2. Edit Posts: Users must have the capability to edit their posts after creation. This includes changing text descriptions and replacing photos if needed.
3. Delete Posts: Users should be able to delete their posts permanently. The deletion process should confirm the action to prevent accidental deletion
   * Upload and display photos:
4. Photo Upload: The application should support photo uploads in various formats (e.g., JPEG, PNG). Users must be able to upload high-quality photos quickly and efficiently.
5. Display Photos: Uploaded photos should be displayed clearly and attractively on the user’s feed and profile. The application should automatically adjust photo sizes to fit the layout while maintaining image quality.

**Interactive Feed:**

* + View, Like, and Comment on Posts:

1. View Posts: Users should see posts from the people they follow in a chronological or algorithmically sorted feed. The feed must be dynamic and reflect the latest posts.
2. Like Posts: Users must be able to like posts, and the total number of likes should be displayed. Liking a post should provide immediate feedback to the user.
3. Comment on Posts: Users should comment on posts, with comments being displayed in real-time. The application should support nested comments to enable threaded conversations.

**Real-time updates**:

* + Immediate Reflections: All user interactions, such as likes, comments, and new posts, must update in real-time. This ensures that users always see the most current information without needing to refresh the page.

**Responsive Design:**

* + **Adaptability:**

1. Device Compatibility: The application should function seamlessly across various devices, including desktops, tablets, and smartphones. It should adapt to different screen sizes and orientations.
2. User Interface (UI): The UI should be designed to provide a consistent and intuitive experience across all devices. Elements should be touch-friendly for mobile users and easily navigable for desktop users.

## Non-Functional Requirements: Non-functional requirements define the system attributes and constraints. These requirements ensure that the application is not only functional but also performant, secure, and user-friendly.

**Performance:**

* Load Time:

1. Fast Loading: The application should load quickly, ideally within 2-3 seconds. This includes the initial load and subsequent page navigations. Minimizing load time enhances the user experience and retention.
2. Concurrent Users: The system should handle a high number of concurrent users without degradation in performance. This includes efficient database queries, optimized API calls, and load balancing.

* Scalability:

1. Scalable Architecture: The application must be designed to scale horizontally to accommodate growing user bases and increased data volume. This includes using cloud services like Firebase that support auto-scaling.

**Security:**

* Data Protection:

1. Encryption: All sensitive user data, including passwords and personal information, must be encrypted both in transit and at rest. Using HTTPS for secure communication and robust encryption algorithms for data storage is essential.

* Access Control:

1. Role-Based Access: Implement role-based access control (RBAC) to ensure that users have appropriate permissions based on their roles. This limits access to sensitive functionalities and data.
2. Audit Logging: Maintain logs of user activities and system events to monitor for suspicious activities and facilitate forensic analysis in case of security incidents.

**Reliability**:

* Uptime:

1. High Availability: The application should have minimal downtime, aiming for an availability of 99.9% or higher. This requires robust infrastructure, failover mechanisms, and regular maintenance.
2. Disaster Recovery: Implement a disaster recovery plan that includes regular backups and a clear strategy for data restoration and service continuity in case of failures.

* Error Handling:

1. Graceful Degradation: In case of errors, the system should fail gracefully, providing meaningful error messages to the user and logging the incidents for troubleshooting.
2. Monitoring: Continuous monitoring of system performance and health using tools like Firebase Analytics and Google Cloud Monitoring to detect and address issues proactively.

**Usability**:

* User Experience (UX):

1. Intuitive Design: The application should be user-friendly, with an intuitive design that requires minimal training. Users should be able to navigate and use features easily, even if they are not tech-savvy.
2. Accessibility: Ensure the application is accessible to users with disabilities. This includes complying with web accessibility standards (e.g., WCAG 2.1) and providing features like screen reader support and keyboard navigation.

* Consistency:

1. Uniform Interface: Maintain a consistent interface design across all pages and functionalities. Consistency helps users feel familiar and comfortable using the application, improving overall satisfaction.

**CHAPTER 4**

**PROJECT PLANNING AND SCHEDULING**

### Pert Chart:

A PERT chart, or Program Evaluation Review Technique chart, is an essential project management tool that aids in the scheduling, organizing, and coordination of tasks within a project. By providing a graphic illustration of a project's timeline and dependencies, it helps project managers identify the critical path, estimate project duration, and manage task sequences effectively.

1. **Understanding the PERT Chart:** A PERT chart is typically presented as a network diagram consisting of numbered nodes representing events or milestones, linked by directional lines (vectors) that denote the tasks. Each task is labeled with pertinent information such as duration, and the direction of the arrows indicates the sequence of tasks.
2. Key Components of a PERT Chart:
3. Nodes: Represent events or milestones. They can be circles or rectangles and are numbered to indicate the sequence.
4. Vectors: Directional lines that connect nodes. They represent tasks and show the flow of activities from one node to the next.
5. Arrows: Indicate the direction of task flow and sequence.
6. Duration Labels: Indicate the estimated time to complete each task.
7. **Steps to Create a PERT Chart:**
8. Identify Tasks: List all the tasks required to complete the project. Each task should be specific and represent a single unit of work.
9. Determine Task Dependencies: Identify the dependencies between tasks. Determine which tasks must be completed before others can begin.
10. Estimate Task Durations: Estimate the time required to complete each task. These estimates should be realistic and based on available resources and historical data.
11. Sequence Tasks: Arrange tasks in the sequence they need to be performed. This involves drawing nodes and connecting them with vectors to show the order of operations.
12. Calculate the Critical Path: Determine the longest path through the PERT chart, known as the critical path. This path indicates the shortest possible project duration and identifies tasks that cannot be delayed without impacting the project schedule.
13. **Example of a PERT Chart for the Social Media Web Application Project:** Here is an example of how a PERT chart can be structured for the Social Media Web Application project, breaking down the major tasks and their dependencies.
14. Nodes and Tasks:
15. Node 1- Project Kickoff.

- Task 1: Initial Planning and Requirements Gathering.

1. Node 2: Design Phase.

- Task 2: UI/UX Design.

1. Node 3: Development Phase

-Task 3: Frontend Development.

-Task 4: Backend Development

-Task 5: Database Setup

1. Node 4: Integration and Testing

-Task 6: Integration of Frontend and Backend.

-Task 7: Testing and Bug Fixing.

1. Node 5: Deployment

-Task 8: Final Deployment and Launch

1. **Benefits of Using a PERT Chart:**

1. Improved Planning and Scheduling: By visualizing the project’s tasks and their interdependencies, a PERT chart helps in accurate planning and scheduling. It ensures that all tasks are accounted for and appropriately sequenced.
2. Enhanced Coordination: The PERT chart facilitates better coordination among team members. It clearly defines the tasks, durations, and dependencies, helping team members understand their roles and responsibilities.
3. Identification of Critical Path: Identifying the critical path is crucial for understanding which tasks directly impact the project’s completion time. This helps in prioritizing tasks and allocating resources effectively.
4. Risk Management: A PERT chart aids in identifying potential bottlenecks and risks in the project. By understanding which tasks are critical, project managers can focus on mitigating risks that could delay these tasks.
5. Effective Monitoring and Control: During the project execution phase, the PERT chart serves as a reference for tracking progress. It helps in monitoring whether tasks are being completed on schedule and provides a basis for making necessary adjustments

The direction of the arrows on the lines indicates the sequence of tasks.

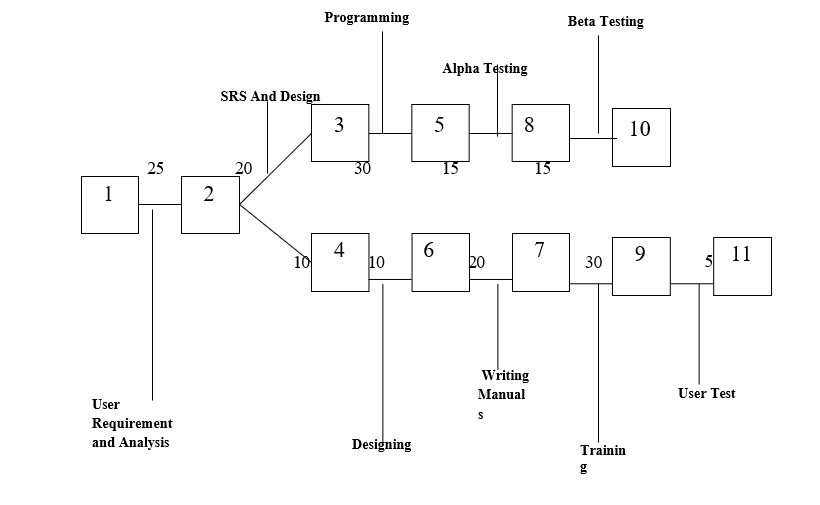


Figure 4.1 Pert Chart

v) **Conclusion:** The PERT chart is a valuable tool for the effective planning and scheduling of the Social Media Web Application project. By providing a visual representation of tasks, dependencies, and timelines, it facilitates better project management, ensuring that the project is completed on time and within scope. Through detailed analysis and strategic planning, the use of a PERT chart enhances coordination, risk management, and overall project control.

**CHAPTER 5**

**HARDWARE & SOFTWARE SPECIFICATION**

### Hardware Specification

### Server:

Processor: Intel Core i5 or equivalent RAM: 8 GB or higher

Storage: 256 GB SSD or higher

### Database Server:

Processor: Intel Core i5 or equivalent RAM: 8 GB or higher

Storage: 256 GB SSD or higher Network Interface: Gigabit Ethernet

**Client Machines:**

Processor: Intel Core i3 or equivalent

RAM: 4 GB or higher

Storage: 128 GB SSD or higher

Network Interface: 100 Mbps Ethernet or Wi-Fi

### Software Specification

### It is developed with the help of React JS, Tailwind CSS, Firebase Firestore, Firebase, Git, GitHub.

### Server-Side Technologies:

### Operating System: Windows Server 2016 or later

### Client-Side Technologies:

Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge Client-Side Scripting: JavaScript

### Development Tool:

Integrated Development Environment (IDE): Visual Studio Code

### Version Control:

Git: Version control for collaborative development

**Security:**

SSL/TLS: Ensure secure data transmission over the network Firewall: Implement firewall rules to restrict unauthorized access

Anti-malware Software: Regularly updated anti-malware software on server and client machines

**CHAPTER 6**

**CHOICE OF TOOLS & TECHNOLOGY**

**React JS**

React JS is a JavaScript library developed by Facebook, primarily used for building user interfaces. It is particularly useful for developing single-page applications where a smooth and responsive user experience is crucial.

1. Key Features:

- Component-Based Architecture: React JS allows developers to build encapsulated components that manage their own state and compose them to create complex UIs.

-Virtual DOM: React JS uses a virtual DOM to optimize and minimize direct manipulations of the actual DOM, resulting in faster performance and a more responsive user interface.

-One-Way Data Binding: This ensures that changes in the model automatically propagate to the view, simplifying data flow and making the application more predictable and easier to debug.

-Reusable Components: React components can be reused across the application, enhancing development efficiency and maintaining consistency in the UI.

**Tailwind CSS**

Tailwind CSS is a utility-first CSS framework that provides low-level utility classes to build custom designs directly in the HTML markup. It helps in creating a modern, responsive, and consistent design.

1. Key Features:

-Utility-First Approach: Tailwind CSS allows developers to apply styles directly in the HTML, resulting in faster development and easier maintenance of styles.

-Responsive Design: Tailwind CSS includes responsive utility variants to ensure designs look good on all screen sizes.

-Customization: It is highly customizable, allowing developers to configure and extend the framework to meet specific project needs.

-Consistency: Using a predefined set of classes ensures a consistent look and feel across the entire application.

**Firebase**

Firebase is a comprehensive app development platform by Google that provides various backend services such as real-time database, authentication, analytics, and more.

1. Key Features:

-Real-Time Database:Firebase’s NoSQL database allows data to be synced across all clients in real-time, ensuring that user interactions and updates are immediately reflected.

-Authentication: Firebase Authentication provides secure user authentication services, including email/password, Google Sign-In, and other social login mechanisms.

- \*\*Hosting:\*\* Firebase Hosting offers fast and secure web hosting for the application, making it easy to deploy and manage web apps.

-Integration: Firebase integrates seamlessly with other Google Cloud services, providing a robust and scalable backend infrastructure.

### Data Flow Diagram

A Data Flow Diagram (DFD) represents the flow of data within a system, illustrating how data is processed by the system in terms of inputs and outputs. It’s an essential tool for the design phase of software engineering, offering a graphical view of the data flow.

In the DFD, four symbols are used and they are as follows:

1. Symbols Used in DFD:
   * 1. Square: Represents a source (originator) or destination of system data.

15

* + 1. Arrow: Identifies data flow, showing data in motion and the pipeline through which information flows.
    2. Circle (Bubble): Represents a process that transforms incoming data flows into outgoing data flows.
    3. Open Rectangle: Denotes a data store, representing data at rest or a temporary repository of data.

### Context Level Diagram

The context level diagram, also known as the zero level DFD, provides an overall context of the system and its operating environment. It shows the whole system as a single process and helps in understanding the system boundaries and interactions with external entities.

1. Context Level Diagram Explanation:

-The context diagram encapsulates the entire system as a single process.

-It identifies external entities interacting with the system, such as users and other systems, and shows the data flows between these entities and the system.

-This high-level diagram is crucial for stakeholders to grasp the system’s scope and boundaries.

**6.3 Level 1 DFD**

Level 1 DFD breaks down the main process into sub-processes, providing a more detailed view of the system’s data flow. This level helps in understanding specific processes and their interactions within the system.

1. Level 1 DFD Explanation:

-The Level 1 DFD decomposes the primary process into its major sub-processes, detailing the internal data flows between these processes.

-It illustrates how data moves between various sub-processes and external entities, offering a clearer view of the system's functionality.

-This diagram is useful for developers and designers to understand the internal workings of the system.

**6.4 Level 2 DFD**

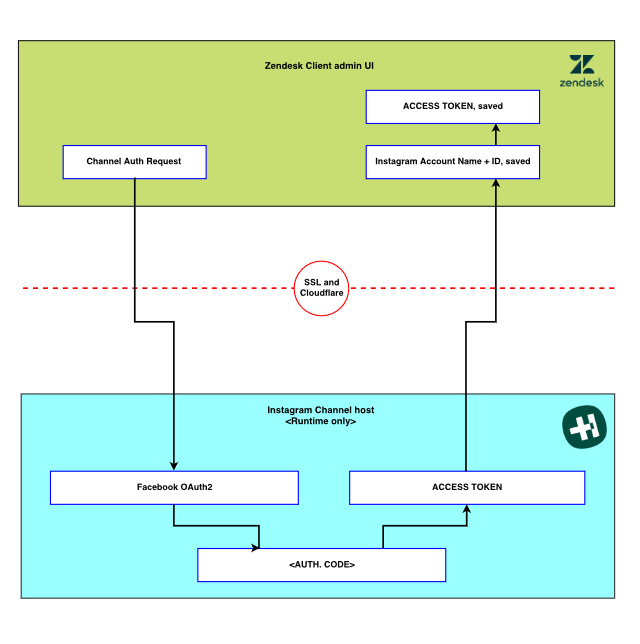
Level 2 DFD offers an even more detailed breakdown of the processes depicted in the Level 1 DFD. It provides a granular view of the system, highlighting specific operations within each sub-process.

1. Level 2 DFD Explanation:

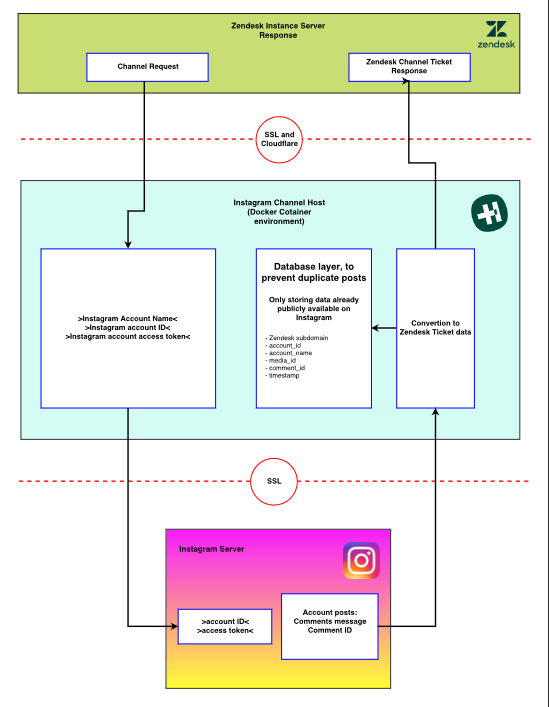
-The Level 2 DFD further decomposes the sub-processes from Level 1 into more detailed components.

-This level of detail helps in identifying specific data flows and storage requirements, ensuring that all aspects of data processing are covered.

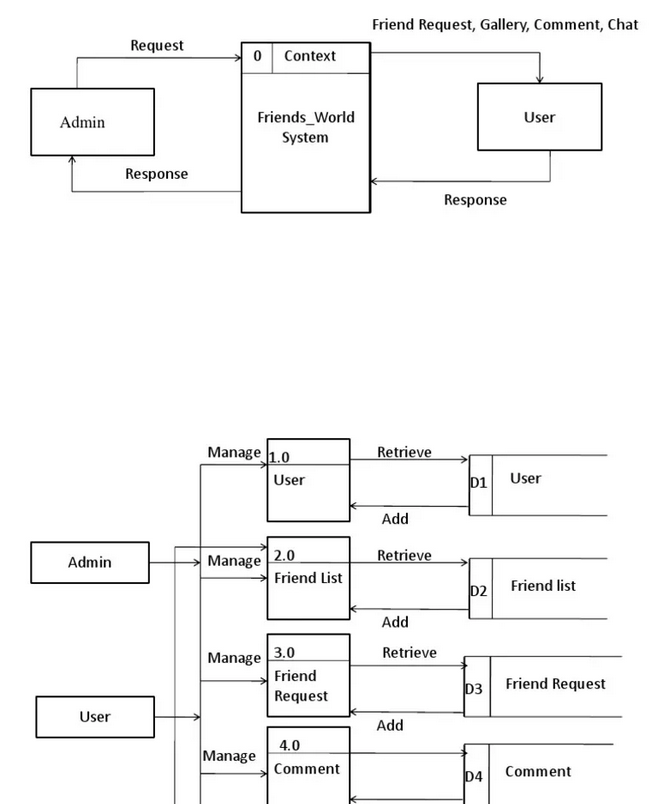
-It is particularly useful during the design and implementation phases to ensure all data handling aspects are properly addressed.



**Fig 6.1** Data Flow Diagram



**Fig 6.2** Level-1 DFD



**Fig 6.3** Level-2 DFD

**Figures**

1. Fig 6.1 Data Flow Diagram (Context Level):

-Represents the entire Social Media Web Application as a single process.

-Shows interactions with external entities such as users and external systems (e.g., third-party services).

1. Fig 6.2 Level-1 DFD:

-Breaks down the main process into major sub-processes such as User Authentication, Post Management, and Real-Time Interaction.

-Illustrates how these sub-processes interact with each other and with external entities.

1. Fig 6.3 Level-2 DFD:

-Provides a detailed view of each sub-process identified in Level 1.

-Details specific operations within each sub-process, such as data validation, storage operations, and user interface interactions.

**Conclusion**

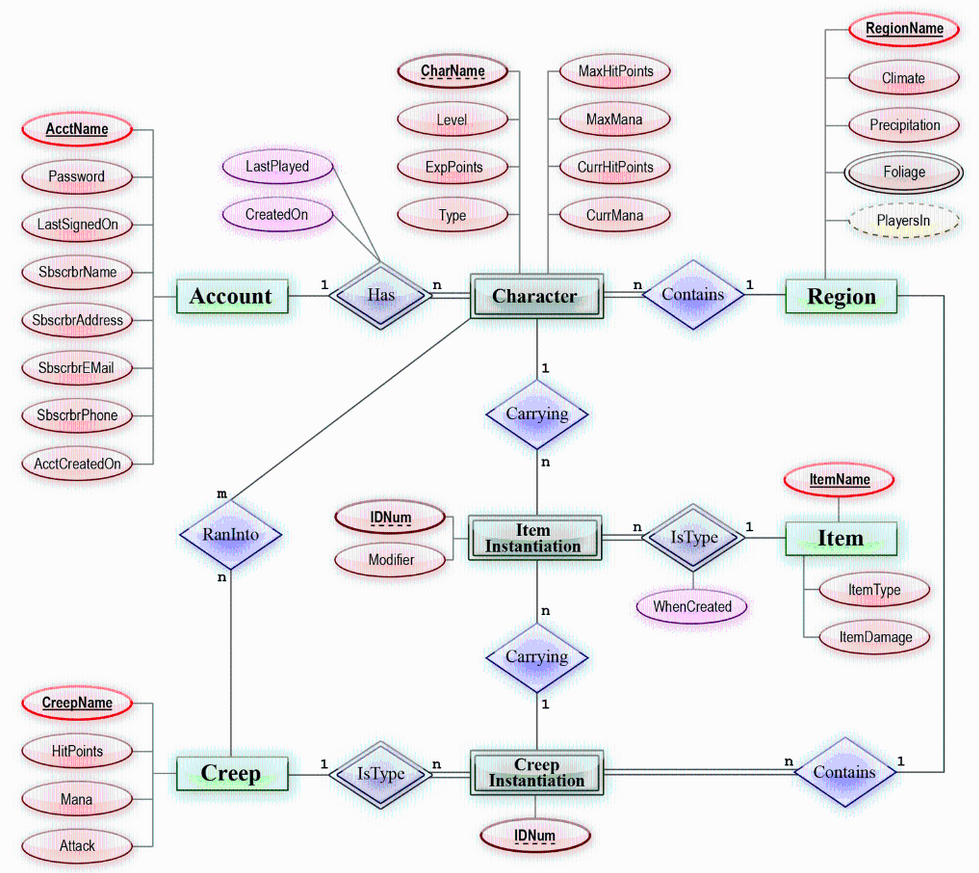
This chapter has provided an in-depth look at the tools and technologies chosen for the Social Media Web Application project, explaining their roles and benefits. Additionally, the chapter detailed the data flow diagrams (DFD), illustrating the flow of data within the system and offering insights into the system’s design and interactions. By leveraging the strengths of React JS, Tailwind CSS, and Firebase, the project aims to deliver a robust, secure, and user-friendly social media platform. The use of data flow diagrams ensures a thorough understanding of data processes, which is critical for effective system design and implementation.

**CHAPTER 7**

**ER-DIAGRAM**

### Entity-relationship model

The entity-relationship model or entity-relationship diagram (ERD) is a data model or diagram for high-level descriptions of conceptual data model, and it provides a graphical notation for representing such data models in the form of entity-relationship diagrams.



**Fig 7.1** E-R Diagram

## 7.2 Class Diagram

## Authentication:

## Classification: Weak Cliet

## Description: Represents users authentication details, including username and password. This class is responsible for users login functionality.

**CHAPTER 8**

**DATABASE**

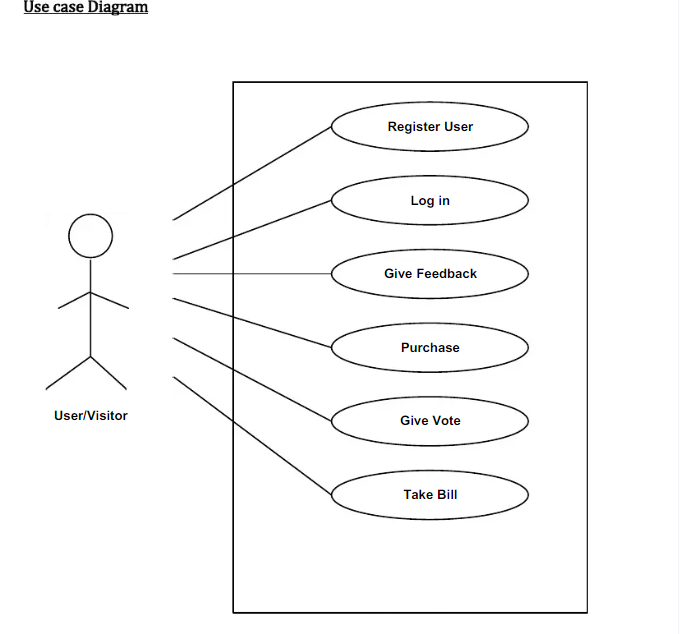
**Use Case Diagram**

A use case diagram is a type of diagram in the Unified Modelling Language (UML) that is used to visualize and describe the functional requirements of a system from an external user's perspective. It provides a high-level view of how users interact with a system and the various functionalities or use cases the system offers in response to those interactions.

Use case diagrams are particularly useful for:

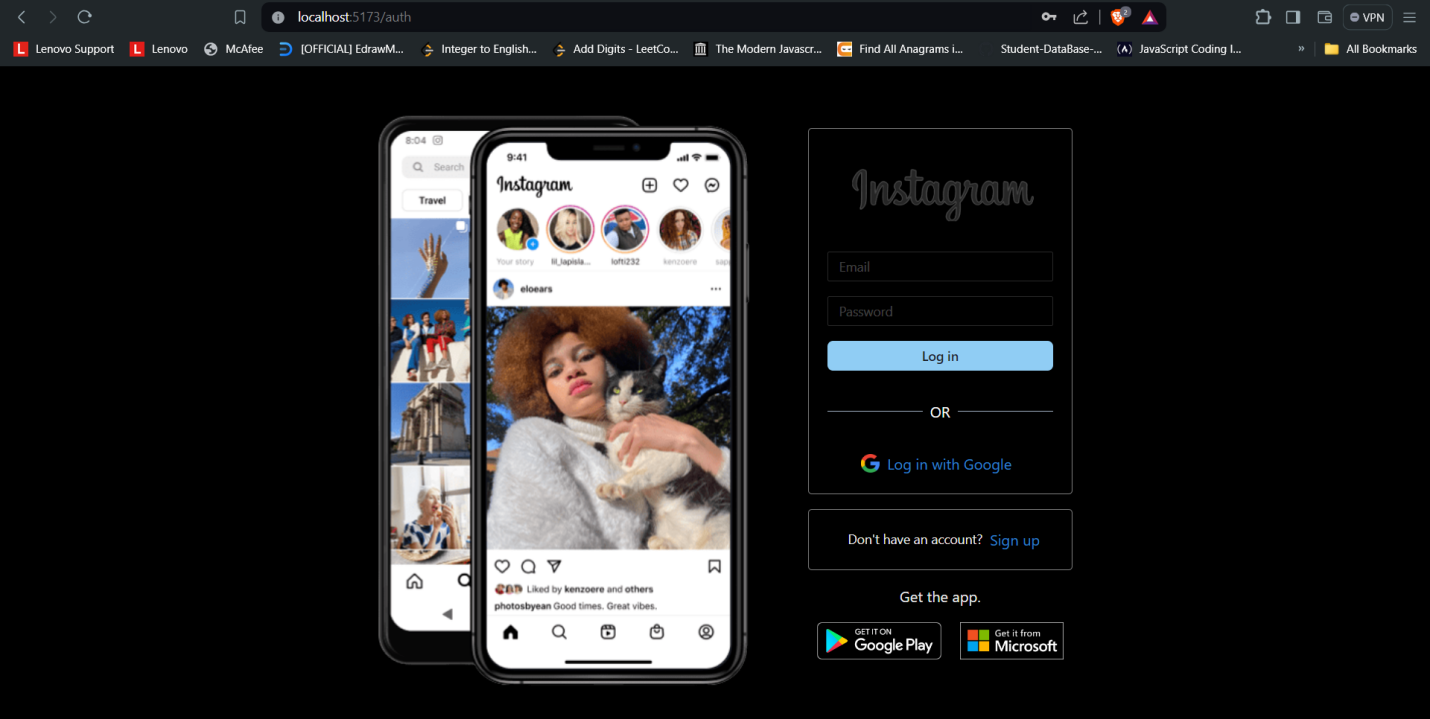
* Communicating the system's functionality and behavior to stakeholders in a visual and understandable way.
* Capturing and documenting high-level user requirements.
* Identifying system boundaries and external interactions.
* Modeling how different use cases relate to each other.

They are a valuable tool in the early stages of software development for understanding and discussing the functional aspects of a system before diving into more detailed design and implementation phases.



**CHAPTER 9**

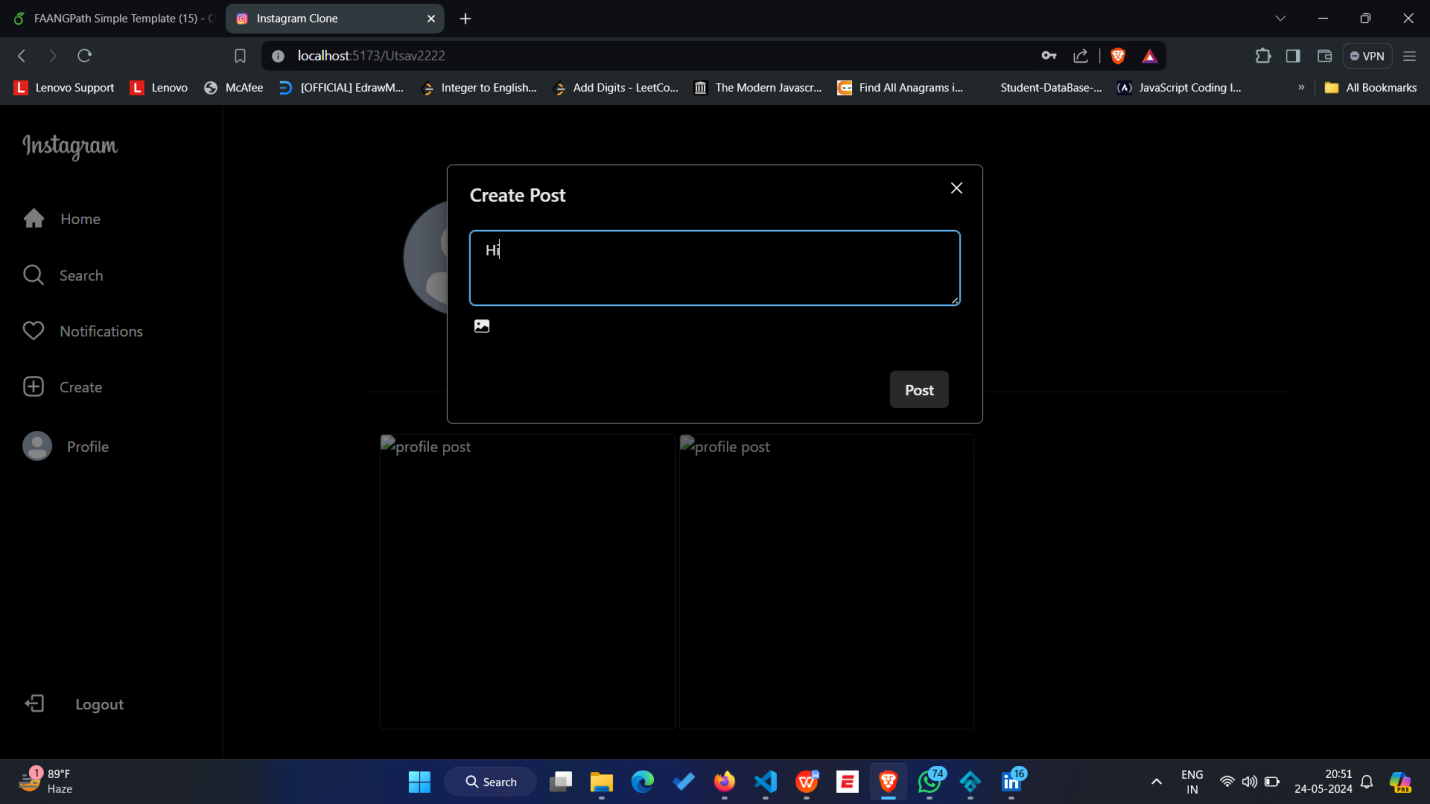
**FORM DESIGN**



**Fig 9.1** Authentication

**About:**

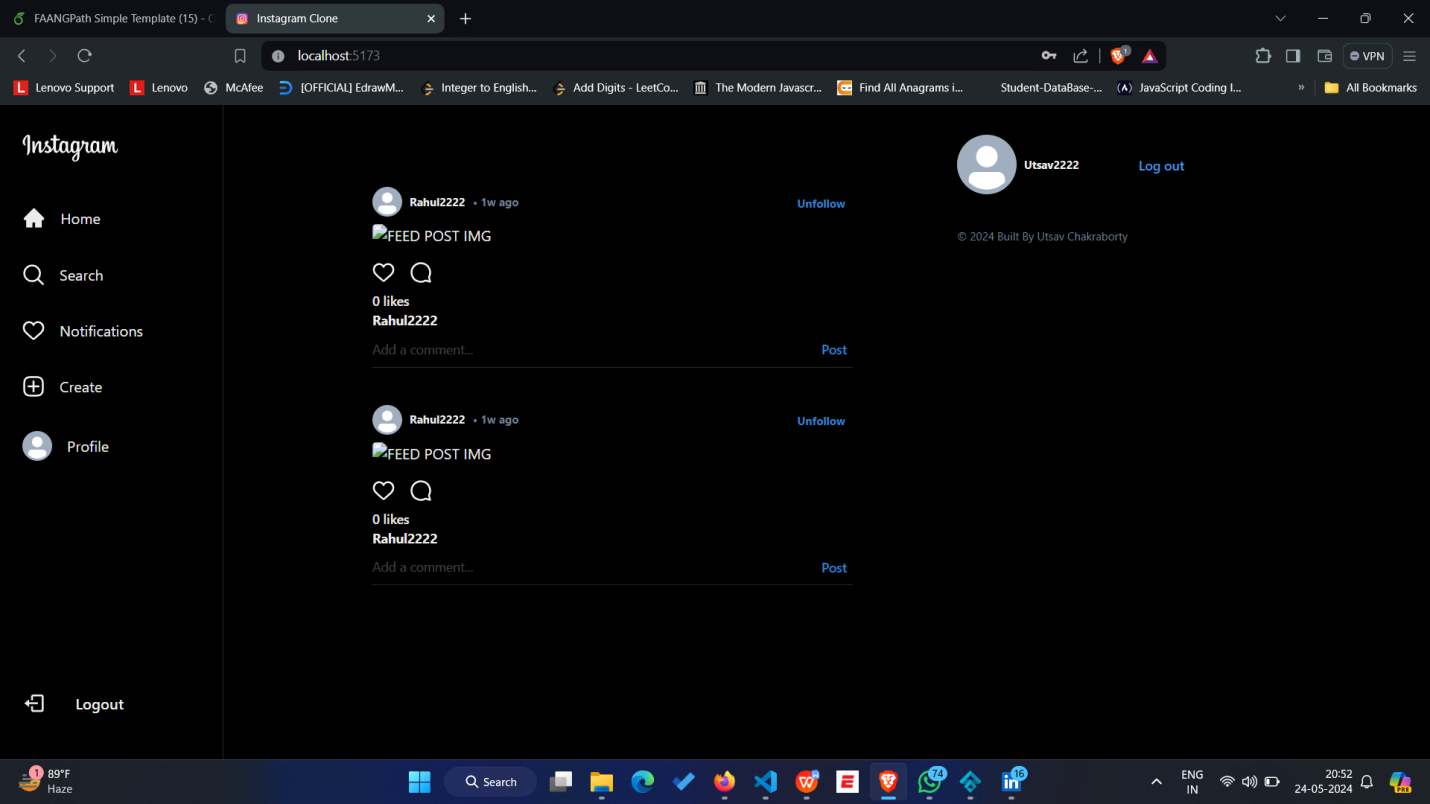
This login form is crucial for the Social Media Web Application, enabling user authentication to secure access to personal profiles and maintain the privacy and security of user interactions on the platform. The login process requires users to enter their credentials, such as a username or email and password, ensuring that only authorized individuals can access their accounts. Advanced security measures like password encryption and multi-factor authentication are implemented to protect user data. This secure login mechanism is essential for safeguarding personal information and maintaining user trust, providing a safe environment for sharing and interacting on the social media platform.



**Fig 9.2** Create Post

**About:**

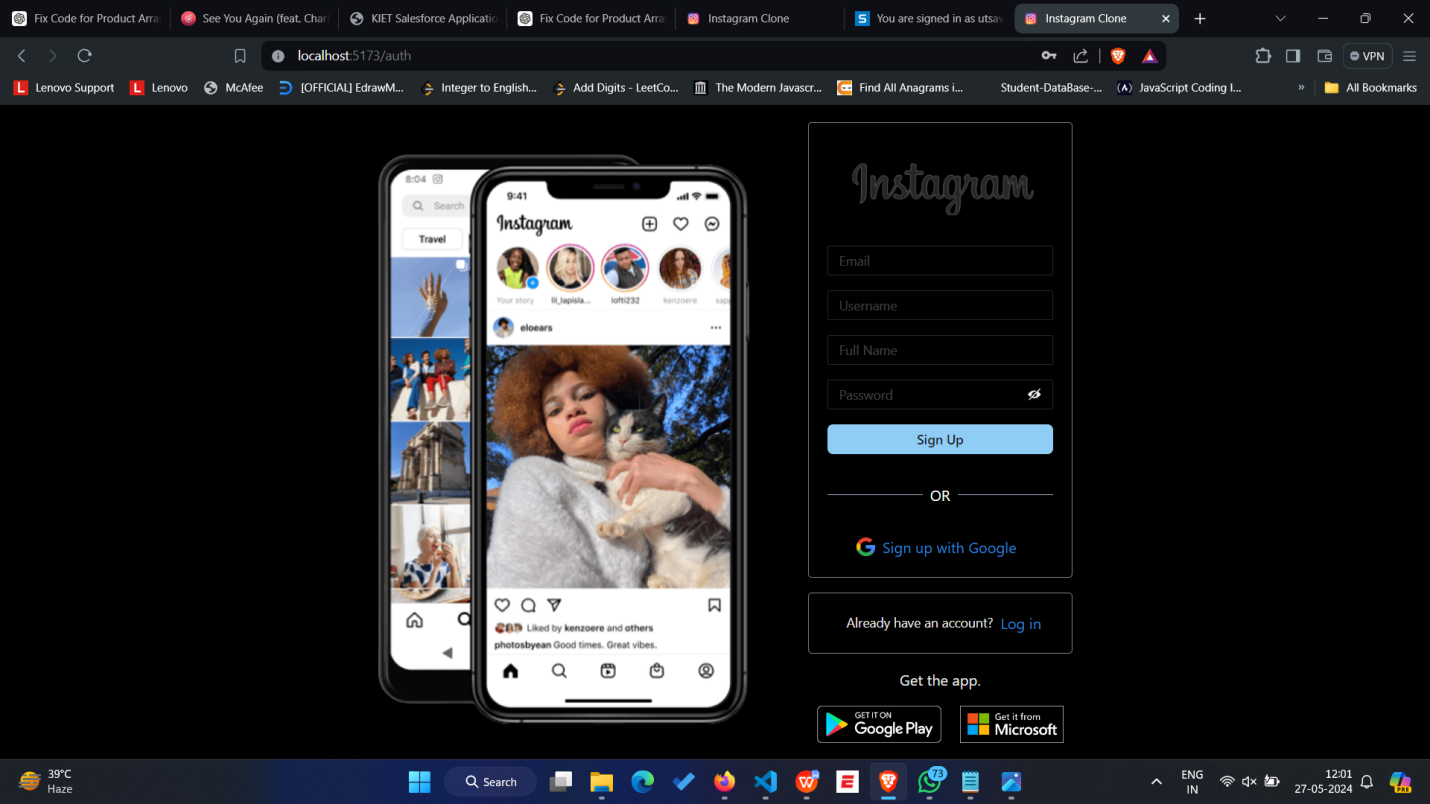
This feature allows users to create and share their posts, enabling them to contribute content, express their thoughts, and engage with others on the Social Media Web Application. Users can upload photos, write captions, and tag friends in their posts. They can also add hashtags to increase the visibility of their content and reach a broader audience. The post creation interface is designed to be user-friendly, with options to edit and preview posts before sharing. Once published, posts appear in the home feed, where followers can like, comment, and share them, fostering interaction and community engagement on the platform.



**Fig 9.3** Home Page

**About:**

The home page displays all the posts from your friends and the people you are following, providing a central hub for viewing and interacting with their latest updates, photos, and activities on the Social Media Web Application. Users can like, comment, and share posts directly from this central feed. Additionally, the home page features personalized content recommendations, trending posts, and highlights from the users you engage with the most. This dynamic feed is continuously updated in real-time, ensuring you never miss out on important moments. The intuitive design and seamless navigation make it easy to stay connected and engaged with your social circle.



**Fig 9.3** Registration Page

**About:**

The registration form is essential for onboarding new users to the Social Media Web Application, allowing them to create personal profiles and join the community. It requires users to provide basic information such as name, email, and password, with additional options to upload a profile picture and write a short bio. The form incorporates validation checks to ensure data accuracy and security, including email verification to prevent fake accounts. The user-friendly design and clear instructions make the registration process quick and straightforward. Once registered, users can start following friends, sharing posts, and engaging with content immediately, fostering a connected and interactive social environment.

**CHAPTER 10**

**TESTING**

### Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionalities of components, sub-assemblies, and/or a finished product it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

### Types of Testing

**Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

### Integration Testing

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using black box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface.

### System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT**

Potential enhancements for the social media web app include implementing direct messaging for user communication, adding stories for sharing temporary updates, and integrating user analytics to track engagement and usage patterns. Future improvements could involve connecting with external services for advanced functionalities, such as third-party content sharing. Additional features might include developing an equalizer for in-app audio customization, lyrics display for music-related posts, offline support for accessibility without internet, and social integration for seamless sharing on other platforms. Stretch goals could involve API integration for enhanced capabilities, user authentication for secure access, collaboration features for group activities, and voice control for hands-free navigation.

**CONCLUSION & REFERNCES**

In conclusion, the Social Media Web App presents an impressive replication of the popular Instagram platform, offering a seamless and enjoyable social networking experience for users. With its user-friendly interface, extensive feature set including photo and video sharing, and personalized feed, it successfully captures the essence of the original service. The app's responsive design ensures accessibility across various devices, enhancing user convenience.

While the Social Media Web App excels in emulating key features of Instagram, it also introduces innovative elements, such as enhanced social integration or unique content discovery algorithms, setting it apart and adding value for users. The development team's dedication to ensuring a smooth and bug-free experience contributes to the overall positive impression of the app.

It is worth noting that continuous updates and improvements will be crucial for the Social Media Web App to stay competitive in the dynamic social networking industry. Regularly adding new features, enhancing security measures, and refining the user experience will be essential to keep users engaged and attract a growing user base.

Ultimately, the Social Media Web App not only serves as a testament to the capabilities of modern technology in replicating successful platforms but also offers a promising alternative for social media enthusiasts seeking a high-quality and innovative networking experience

References

<https://reactjs.org/docs/getting-started.html>

<https://firebase.google.com/docs>

[https://tailwindcss.com/docs](https://tailwindcss.com/docs" \t "https://chatgpt.com/c/_new)

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide" \t "https://chatgpt.com/c/_new)

[https://developer.mozilla.org/en-US/docs/Web/HTML](https://developer.mozilla.org/en-US/docs/Web/HTML" \t "https://chatgpt.com/c/_new)

[https://developer.mozilla.org/en-US/docs/Web/CSS](https://developer.mozilla.org/en-US/docs/Web/CSS" \t "https://chatgpt.com/c/_new)

<https://www.w3schools.com/css/css_rwd_intro.asp>

<https://www.atlassian.com/agile/project-management>