

# **Social Media Platform** **Project Report**

**By**  
**Danish Shaikh –AF04954641**  
**Tanvesh Vaity – AF04955993**

# Index

Sr.no	Topic	Page no
1	Title of Project	1
2	Acknowledgement	3
3	Abstract	4
4	Introduction	5
5	Objective	6
6	System Analysis	7
7	<u>System Design</u>	22
8	Screenshots	32
9	Coding	39
10	Testing	46
11	Report	47
12	Future Scope	48
13	Conclusion	48
14	Bibliography	49
15	References	49

## **Acknowledgement**

The project “**Social Media Platform**” is the Project work carried out by

<b>Name</b>	<b>Enrollment No</b>
<b>Danish Shaikh</b>	<b>AF04954641</b>
<b>Tanvesh Vaity</b>	<b>AF04955993</b>

Under the Guidance.

We are sincerely thankful to our project guide, **Rajshri Thete Ma’am**, for her valuable guidance and continuous support in completing our project His suggestions and valuable information regarding the formation of the Project Report have provided me a lot of help in completing the Project and its related topics.

We are also thankful to my family member and friends who were always there to provide support and moral boost up.

## **Abstract**

The project called **Social Media Platform** is about creating a website where users can connect, share, and interact with each other in a simple and engaging way. This system replaces the traditional offline or limited communication methods with a digital platform where users can easily create posts, like, comment, and share updates in real time. It also stores user content securely, making it easier to manage and access anytime.

This platform lets users connect through profiles, follow others, browse content by category or interest, and engage with posts through likes, comments, and shares. It also includes features like comment analysis, showing whether feedback is positive, neutral, or negative, helping users and administrators understand audience engagement better.

The system is designed to be user-friendly, responsive on mobile devices, and secure. It allows real-time updates, content moderation, and smart analytics to track engagement trends. It helps communities and organizations build stronger connections, save time, and reach a larger audience. This platform focuses on interaction, accessibility, and data-driven insights to create a meaningful social experience.

# 1. Introduction

In today's digital age, social media has become one of the most powerful tools for communication, information sharing, and community building. Traditional methods of interaction and content sharing are limited in speed, reach, and accessibility. To address these challenges, this project introduces a **Social Media Platform** that provides users with a modern, interactive, and user-friendly environment to connect, share, and engage with one another.

The platform enables users to create personal profiles, post updates, share multimedia content, and interact with others through likes, comments, and shares. It categorizes posts based on interests or topics, making it easier for users to discover relevant content. A key feature of the system is **real-time updates and comment analysis**, which classifies feedback as positive, negative, or neutral. This not only enhances user interaction but also provides administrators with valuable insights into public opinion and engagement trends.

Designed with simplicity and accessibility in mind, the system is mobile-responsive, secure, and easy to use. It also includes moderation tools to maintain a safe and respectful environment for all users. By combining social interaction with intelligent analysis, this platform aims to create a dynamic digital community that promotes meaningful engagement, fosters connections, and empowers organizations and individuals to reach wider audiences efficiently.

## **1.1 Objective of the Present Work**

The objectives of this project are as follows:

- To develop a web application for an Social Media Platform so they can connect with others.
- To allow users to sign up, log in, and maintain secure sessions.
- Enable users to update their profile details and manage personal information.
- Allow users to create, edit, delete, and view posts.
- To provide engagement options such as likes, comments, and shares for better user interaction.
- To Store user data, posts, likes, and comments in a structured and relational manner using MySQL.

# **System analysis**

## **PROBLEM DEFINITION**

In today's world, social media has become an essential tool for communication, self-expression, and information sharing. However, many existing platforms are either overly complex, resource-intensive, or lack transparency and control for users. Traditional systems of communication do not provide real-time interaction, organized data management, or wide accessibility.

The absence of a simple, customizable, and lightweight platform makes it difficult for smaller communities, organizations, and learners to build their own social networking solutions

### **3.1 Preliminary Investigation**

#### **Purpose**

The purpose of the Social Media Platform is to provide a modern, secure, and interactive digital space where users can easily connect, share content, and engage with one another. This system is designed to overcome the limitations of traditional communication methods and existing complex platforms by offering a lightweight, customizable, and scalable solution.

#### **Benefits**

The portal provides several advantages:

- Real-Time Interaction – Users can instantly share posts, like, and comment, making communication faster and more engaging.
- Secure & Reliable – With authentication and MySQL database storage, user data is protected and managed efficiently.
- User-Friendly & Responsive – The React-based interface ensures a smooth experience across desktop and mobile devices
- Scalable & Expandable – The Node.js + MySQL backend provides flexibility to handle more users and supports future feature upgrades.

## Proposed System

The proposed system offers a structured, secure, and interactive approach to social networking:

- Users can create accounts, share posts, like and comment on content, and manage their profiles.
- Create, edit, and delete posts with the ability to share text, images, or other media.
- Engage with posts through likes and comments in real time.
- Manage personal profiles, including updating details and viewing activity.
- Browse a news feed to stay updated with posts from other users.

## 3.2 Feasibility Study

The feasibility study evaluates whether the **Social Media Platform** project is practical, achievable, and beneficial. This assessment ensures the system can be successfully implemented within available resources, technology, and constraints.

### Types of Feasibility Analysis

- **Technical Feasibility**
  - ✓ The platform is built using **React.js** for the frontend and **Node.js with Express.js** for the backend, ensuring compatibility with modern web technologies.
  - ✓ **MySQL relational database** securely manages user profiles, posts, likes, and comments..
  - ✓ The system supports **RESTful APIs** for efficient communication between frontend and backend.
  - ✓ The application is designed to be responsive and scalable, ensuring smooth performance for a growing user base.
- **Economic Feasibility**
  - ✓ The project relies on **open-source technologies** such as React, Node.js, and MySQL, reducing overall development costs.
  - ✓ Automated features like post management and secure authentication minimize manual intervention.



- **Operational Feasibility**

- ✓ Users can easily **sign up, log in, create posts, like, and comment**, ensuring high usability
- ✓ **Profile management** and an intuitive posts provide a seamless user experience.

- **Schedule Feasibility**

- ✓ The development timeline is practical, covering **UI design, backend API development, database integration, testing, and deployment**.

- ✓ **Modular development** allows independent implementation of features like authentication, posts, and comments, improving flexibility.

- **Social Feasibility**

- ✓ The project encourages **healthy interaction and community building** among users.
- ✓ Features like commenting and liking foster **positive engagement** and collaboration.
- ✓ The platform is **accessible to all users**, regardless of technical expertise, thanks to its simple and responsive design.

### [3.3 Project Planning](#)

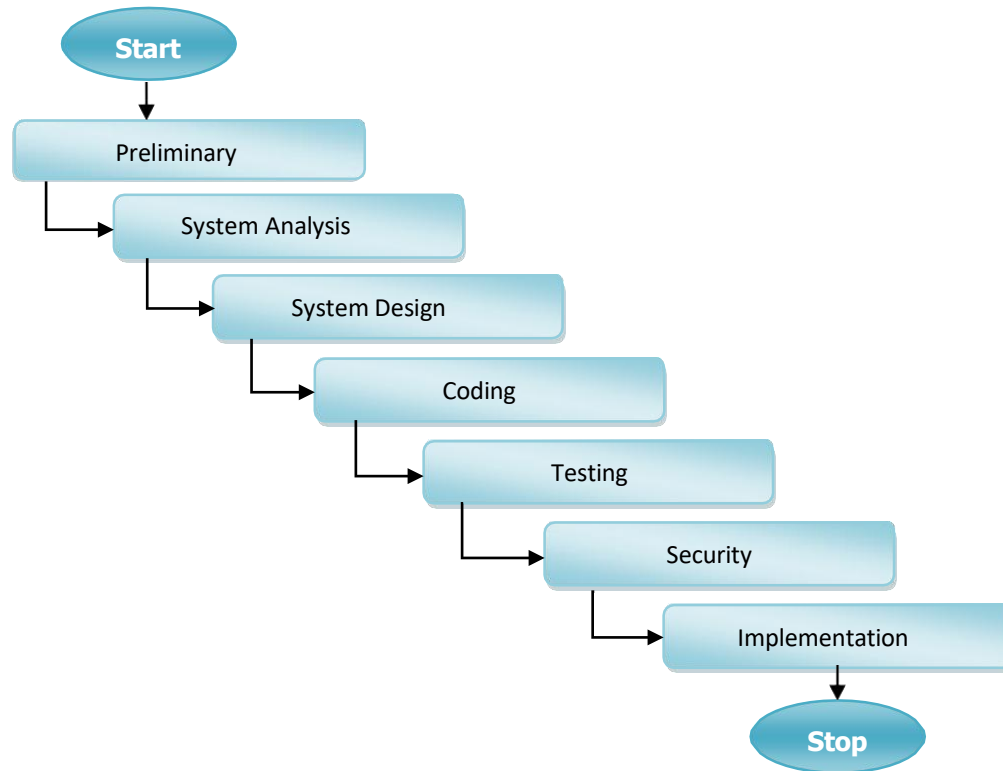
#### [Purpose of Project Planning](#)

Project planning ensures that the development of the **Social Media Platform** follows a structured approach. It helps define the workflow, resource allocation, timelines, and risks involved to deliver the system efficiently.

#### [Phases Covered in the Plan](#)

The planning is divided into different phases to ensure smooth execution:

1. **Preliminary Investigation** – Understanding the project scope and objectives.
2. **System Analysis** – Identifying challenges, gathering requirements, and defining solutions.
3. **System Design** – Structuring modules, database design, and UI development.
4. **Coding** – Developing the platform using Reactjs and express.js and integrating MySQL
5. **Security** – Implementing authentication, data encryption, and user privacy measures.
6. **Testing** – Performing unit testing, integration testing, and user acceptance testing.
7. **Implementation** – Deploying the final system and ensuring smooth operation.



### 3.4 Software Requirement Specification (SRS)

The **Software Requirement Specification (SRS)** outlines the fundamental requirements of the **Online News Portal with Sentiment Analysis** to ensure efficient functionality, usability, and maintainability.

#### System Overview

The Online News Portal allows users to access and interact with news articles across multiple categories. The system is structured into **three modules**:

1. **User Module** – Enables users to browse news, search for articles, and post comments.
2. **Admin Module** – Provides tools for news management, user moderation, and sentiment analysis.
3. **Sub-Admin Module** – Allows sub-admins to manage articles with limited access rights.

## Software & Hardware Requirements

### Software Requirements

- **Frontend:** HTML, CSS, Reactjs
- **Backend:** Express.js for api and data handling
- **Database:** MySQL for storing posts, comments, and user details

### Hardware Requirements

- **Processor:** Intel i5 or higher
- **RAM:** Minimum 8GB
- **Storage:** At least 100GB for database and media files
- **Connectivity:** Internet access for real-time updates

## 3.5 Functional Requirements

### 1. User Module

Users can:

- Register & Login – Create accounts, log in securely, and manage their personal profiles
- Post Management – Create, edit, and delete their own posts.
- Interaction Features – Like and comment on posts from themselves and other users.
- Profile Viewing – View and update their own profile details and browse profiles of other users.
- News Feed – Access a real-time feed displaying posts from followed users or the community.
- Search & Explore – Search for specific posts or users using keywords.

### 2. Admin Module

Admins have full control over the platform and can:

- Secure Login System – Access the admin dashboard through authentication.
- User Management – Add, suspend, or delete users to maintain a safe environment.
- Content Moderation – Monitor posts and comments, removing inappropriate or harmful content.
- Dashboard Management – Access analytics such as number of users, posts, and interactions.

### 3.6 Software Engineering Paradigm

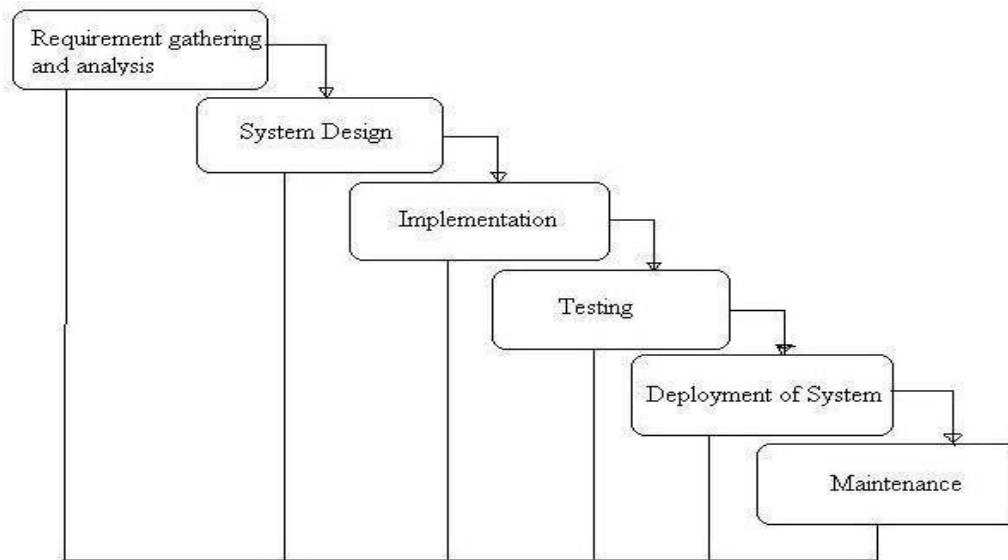
The development of the **Social Media Platform** follows a structured approach to ensure efficiency, reliability, and maintainability. The chosen paradigm helps streamline the project by defining clear phases while allowing iterative improvements.

#### Development Model: Adapted Waterfall Model

The **Waterfall Model** is traditionally a linear approach, but for this project, an **iterative feedback mechanism** is incorporated. This helps refine earlier phases based on insights gathered during implementation.

#### Key Adaptations in the Waterfall Model:

1. **Structured Phase Progression** – Each stage follows a defined sequence, ensuring clarity in execution.
2. **Iterative Refinements** – Feedback loops allow adjustments, especially between testing and coding.
3. **Defined Milestones** – Each stage reaches completion before moving to the next phase.
4. **Flexible Adjustments** – Overlapping is permitted when necessary to enhance efficiency.



## Phases of Development

### 1. Requirement Analysis & System Study

- Identifying project goals, challenges, and functional specifications.
- Gathering requirements for user profiles, posts, likes, comments, and admin functionalities.

### 2. System Design

- Structuring the database (MySQL) with entities like users, posts, likes, and comments.
- Designing RESTful APIs for interaction between frontend and backend.
- Creating wireframes and UI mockups for a responsive React-based interface.

### 3. Implementation (Coding)

- Frontend Development: React.js for building user interfaces with responsive design.
- Backend Development: Node.js with Express.js for API development and server-side logic.
- Database Integration: MySQL for managing user data, posts, and interactions.
- Authentication & Security: Implementing JWT/bcrypt for secure user login and sessions.

### 4. Testing & Debugging

- Testing security of website
- Debugging for performance improvements.

### 3.7 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 can be manual, automated, or a combination of both.


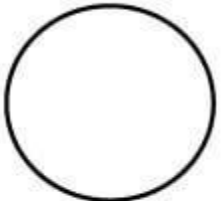

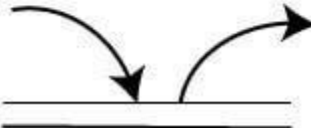
It shows how data enter and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

**The following observations about DFDs are essential:**

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows in a flow chart represent the order of events; arrows in DFD represent flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represent decision points with multiple existing paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.

Standard symbols for DFDs are derived from the electric circuit diagram analysis and are shown in fig:

Symbol	Name	Function
	Data flow	Used to Connect Processes to each other, to sources or Sinks; the arrow head indicates direction of data flow.
	Process	Performs Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

### Symbols for Data Flow Diagrams

**Circle:** A circle (bubble) shows a process that transforms data inputs into data outputs.

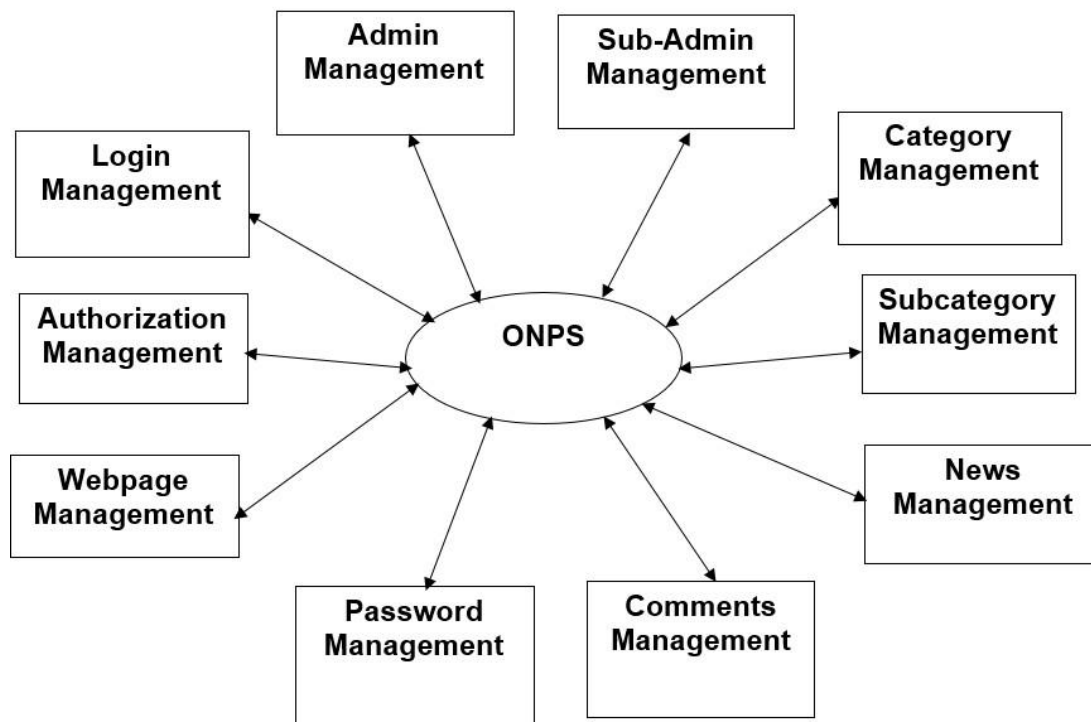
**Data Flow:** A curved line shows the flow of data into or out of a process or data store.

**Data Store:** A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements.

**Source or Sink:** Source or Sink is an external entity and acts as a source of system inputs or sink of system outputs.

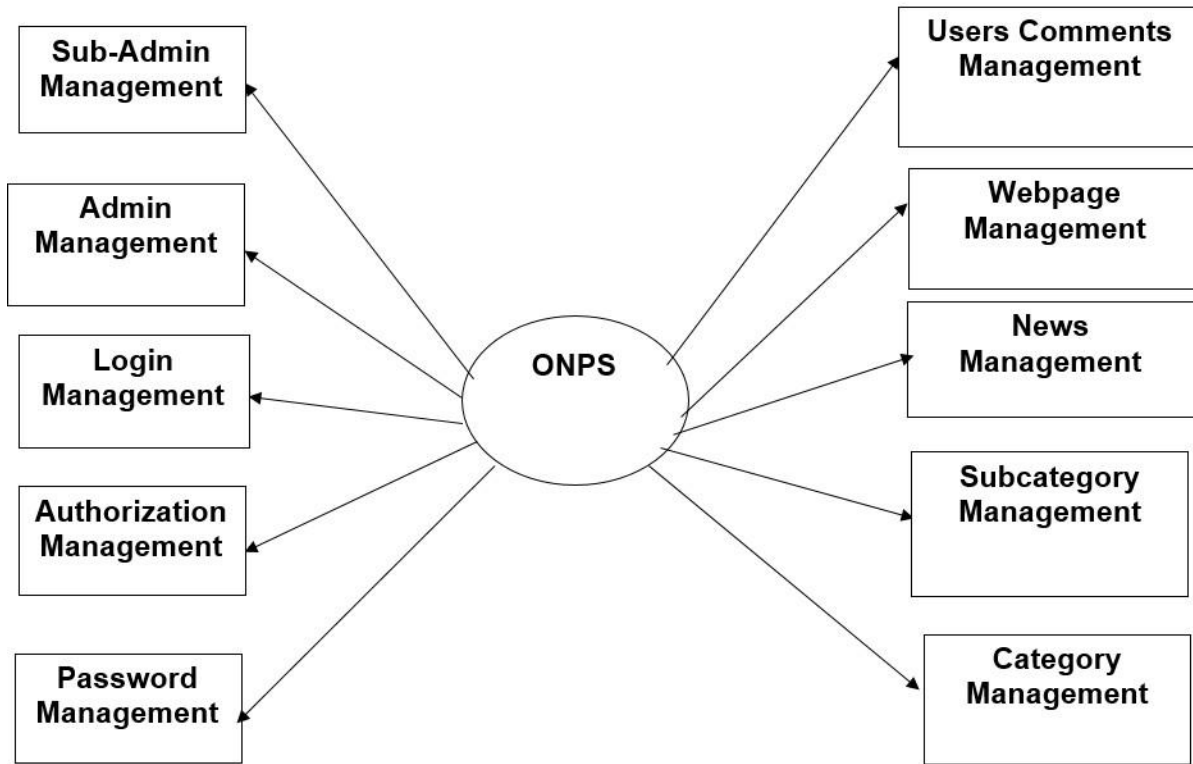


### Zero Level DFD



The Zero-Level DFD of the Social Media Platform shows the main processes interacting with the system, such as **Admin, Users, Posts, Comments, Likes, Followers/Following, and Media Management**. It highlights the flow of data between the central system and these modules, representing how each core function is managed within the platform..

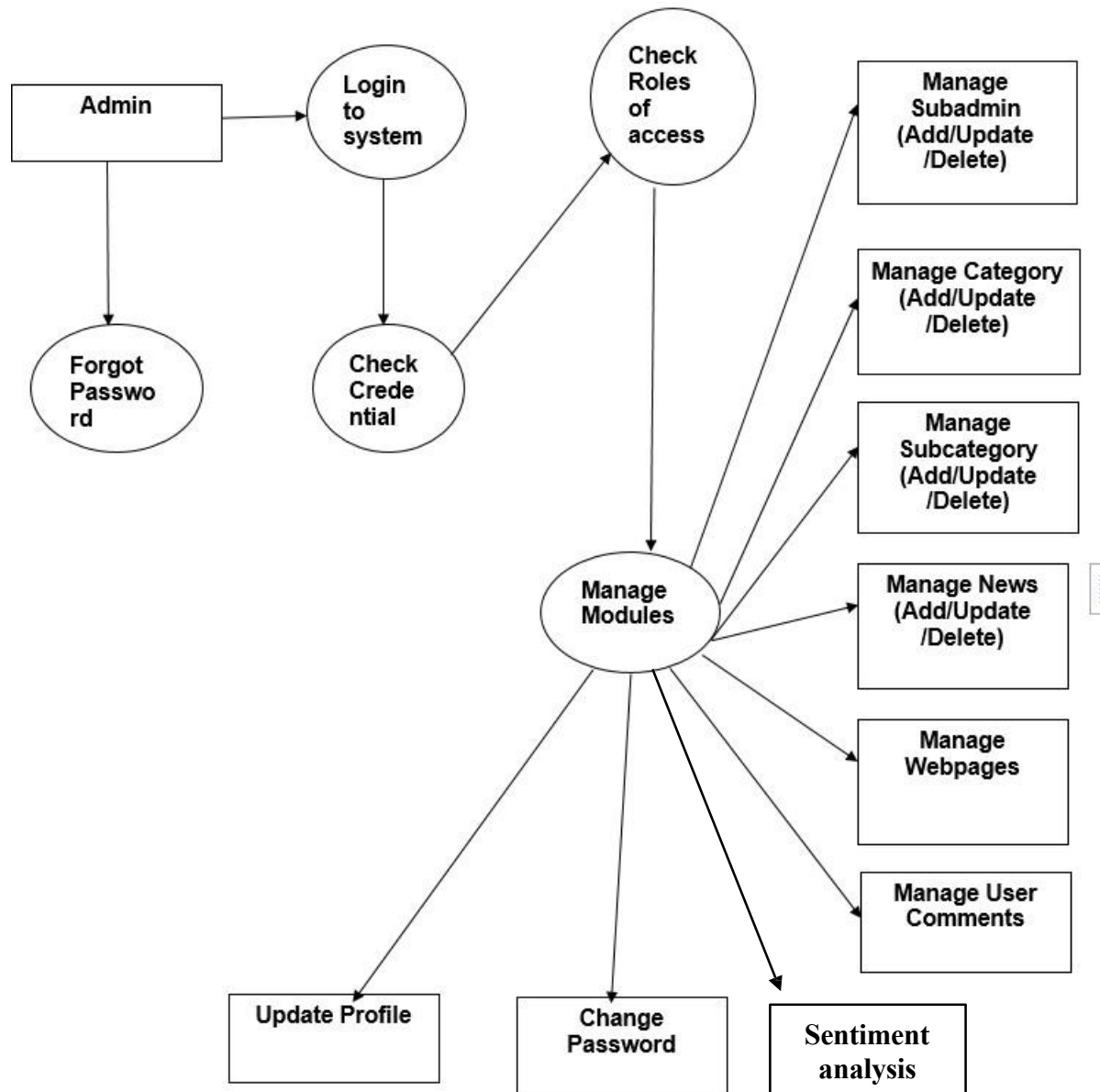
### First Level DFD



### First-Level DFD:

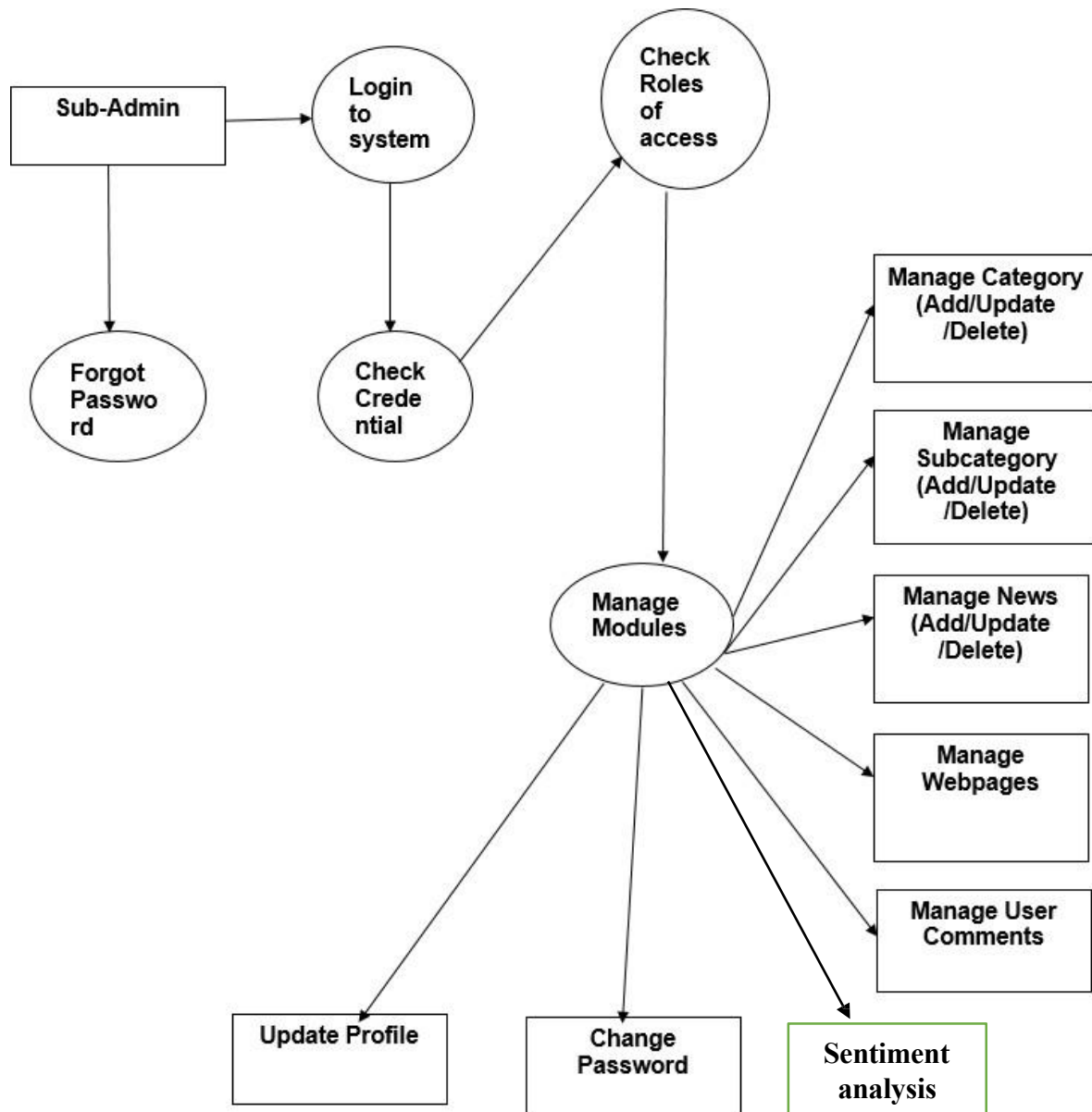
The First-Level DFD illustrates how the system handles key functions like **user registration/login, profile management, and admin authorization**, and connects them to content-related modules such as posts, comments, likes, stories, and follower/following management to ensure smooth platform operations.

### Second Level DFD



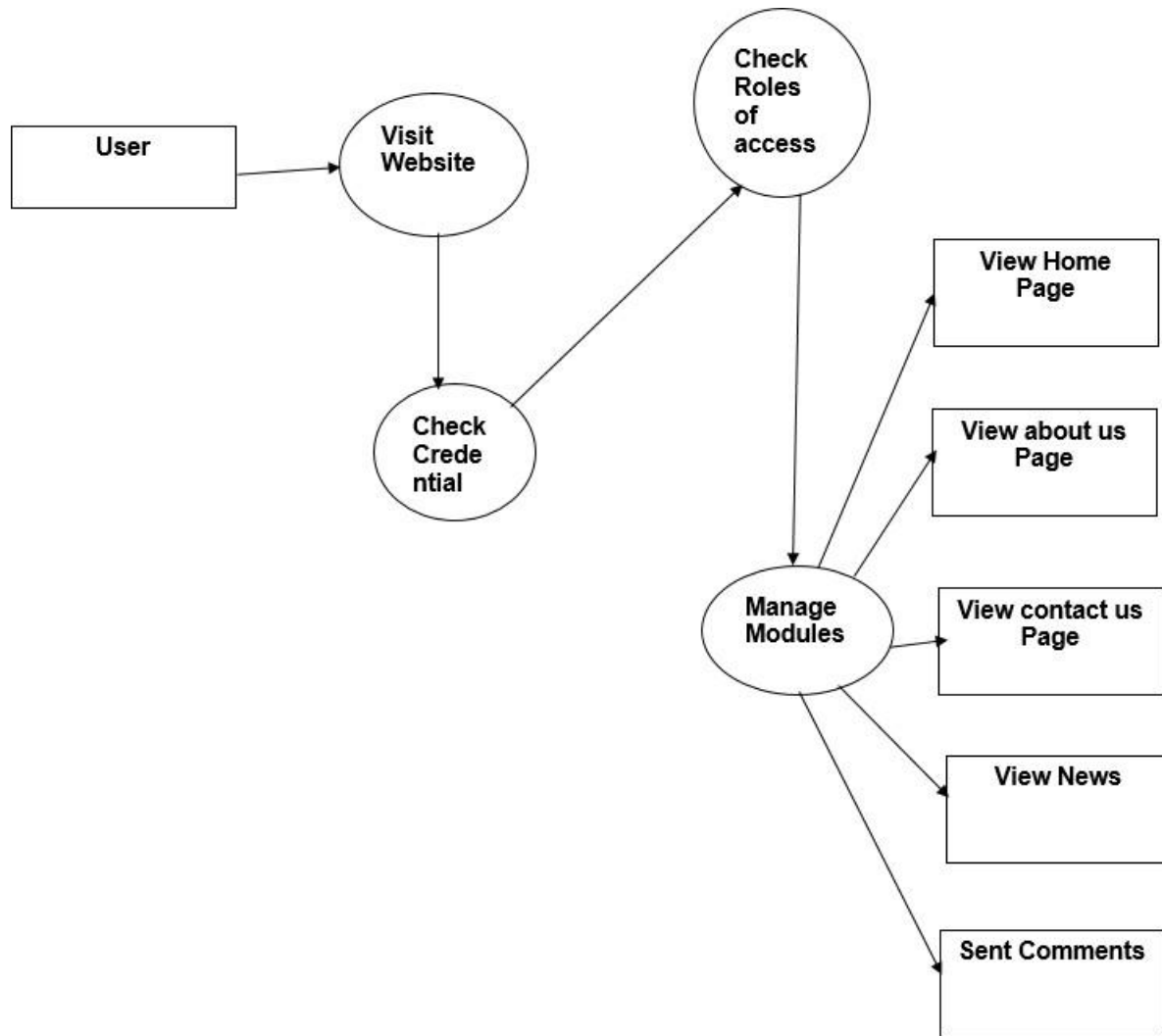
#### Second-Level DFD (Admin):

The Second-Level DFD shows the internal working of the **Admin** in the Social Media Platform. It includes processes like login, credential verification, role-based access, and module management. The Admin can manage users, moderate posts and comments, access analytics, and update their profile and password.



### Second-Level DFD (Sub-Admin/Moderator):

This DFD represents the internal workflow for a **Sub-Admin or Moderator**. After logging in and verifying credentials, the system checks the user's access role and allows them to moderate content such as posts, comments, and media. Additionally, they can update their profile and password.

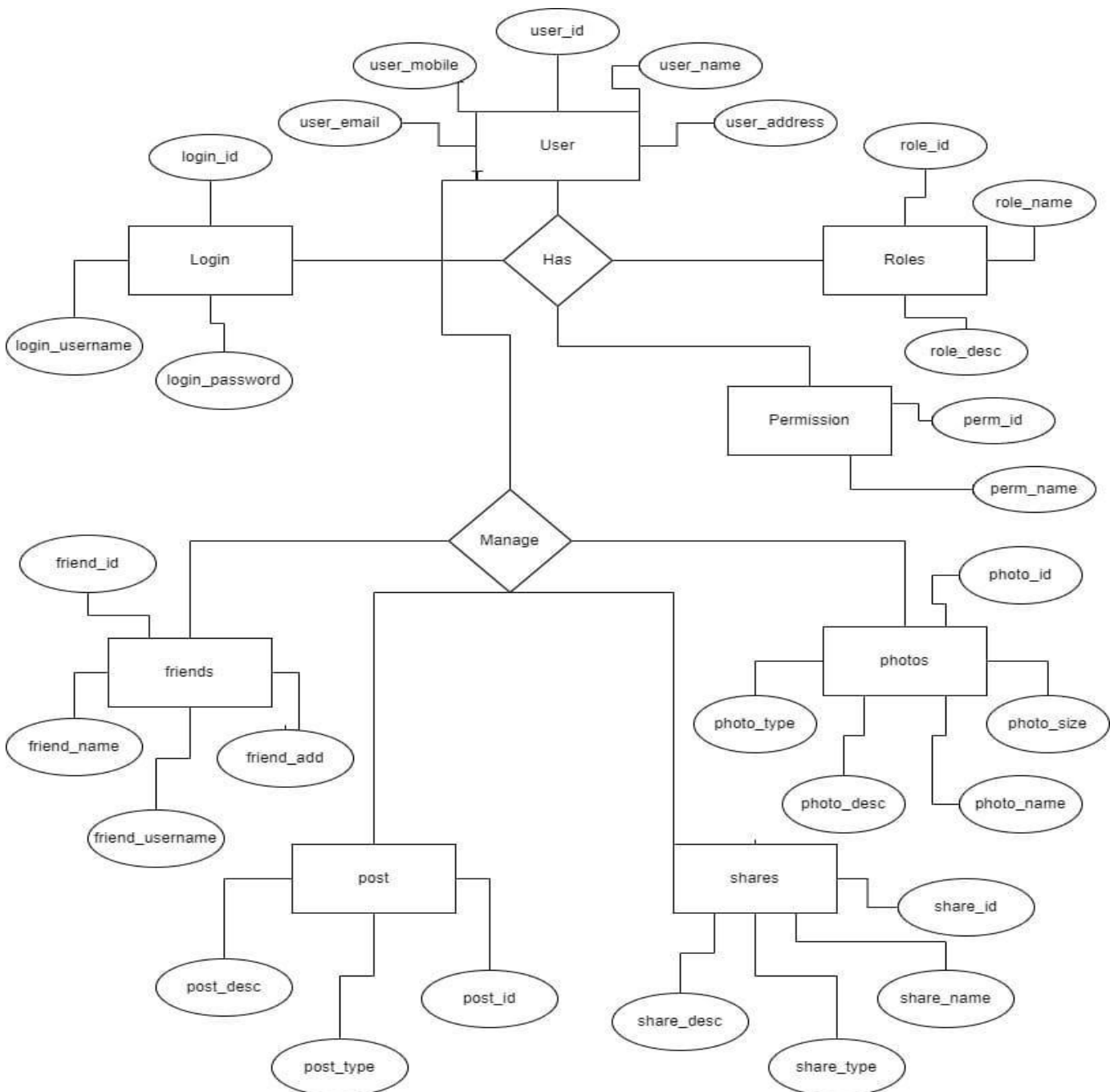


### Second-Level DFD (User):

This DFD illustrates the **User's interaction** with the Social Media Platform. Users can register, log in, update their profiles, create posts, like and comment on content, follow other users, and view a personalized feed of posts from their network or the broader community.

## ER diagram

The ER Diagram of the Social Media Platform illustrates the key relationships between **Users, Posts, Comments, Likes, Followers/Following, and Media Content**, ensuring structured data management and seamless user interaction. It captures how users create posts, interact through comments and likes, follow other users, and share media, while providing a clear framework for data organization, privacy control, and content moderation. This structure supports efficient querying, personalized feeds, and analytic insights to enhance user engagement and platform performance.



## 4. System design

### 4.1. Module

Users can:

- Register & Login – Create accounts, log in securely, and manage their personal profiles
- Post Management – Create, edit, and delete their own posts.
- Interaction Features – Like and comment on posts from themselves and other users.
- Profile Viewing – View and update their own profile details and browse profiles of other users.
- News Feed – Access a real-time feed displaying posts from followed users or the community.
- Search & Explore – Search for specific posts or users using keywords.

### Admin Module

Admins have full control over the platform and can:

- Secure Login System – Access the admin dashboard through authentication.
- User Management – Add, suspend, or delete users to maintain a safe environment.
- Content Moderation – Monitor posts and comments, removing inappropriate or harmful content.
- Dashboard Management – Access analytics such as number of users, posts, and interactions.

## 4.2 DATA STRUCTURE OF ALL MODULES:

We have organized one database **news portal** for system design. It can be accessed directly or sequentially by registered. The database determines files, record, fields, and characters. It can be easily controlled and updated. **“Social Media Platform”** contains **15 MySQL tables**(In this MySQL 6 table is customized and 9 table made by default in django) :

### Customized Tables Details

#### Admin and Sub-Admin Table:(user)

This store admin personal and login details.

```
mysql> describe users;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
username	varchar(45)	NO		NULL	
email	varchar(45)	NO		NULL	
password	varchar(200)	NO		NULL	
name	varchar(45)	NO		NULL	
coverPic	varchar(200)	YES		NULL	
profilePic	varchar(200)	YES		NULL	
city	varchar(45)	YES		NULL	
website	varchar(45)	YES		NULL	

```
9 rows in set (0.00 sec)

mysql>
```

#### User Posts Table: Table name( social media Posts)

This table store the details User posts.

```
mysql> describe posts;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
desc	varchar(200)	YES		NULL	
img	varchar(200)	YES		NULL	
userId	int	NO	MUL	NULL	
createdAt	datetime	YES		NULL	
location	varchar(100)	YES		NULL	

```
6 rows in set (0.00 sec)
```



### User Story Table: Table name(social media Story)

This table store stories of users

```
mysql> describe stories;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int           | NO   | PRI | NULL    | auto_increment |
| img   | varchar(200)  | NO   |     | NULL    |                |
| userId | int          | NO   | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

### Comments Table: Table name( social media post comments)

This table store the comments details which is given by different users or followers.

```
mysql> describe comments;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int           | NO   | PRI | NULL    | auto_increment |
| desc       | varchar(200)  | NO   |     | NULL    |                |
| createdAt  | datetime      | YES  |     | NULL    |                |
| userId     | int           | NO   | MUL | NULL    |                |
| postId     | int           | NO   | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

**Likes Table: Table name( social media post likes)**

This table store the details of social media post Liked by the users

```
mysql> describe Likes;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
userId	int	NO	MUL	NULL	
postId	int	NO	MUL	NULL	

```
3 rows in set (0.00 sec)
```

```
mysql>
```

**Relationship Table: Table name(social media Relationships)** relationships table stores the mapping of followers and followings between users. Each record represents one user (followerUserId) following another user (followedUserId).

```
mysql> describe relationships;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
followerUserId	int	NO	MUL	NULL	
followedUserId	int	NO	MUL	NULL	

```
3 rows in set (0.00 sec)
```

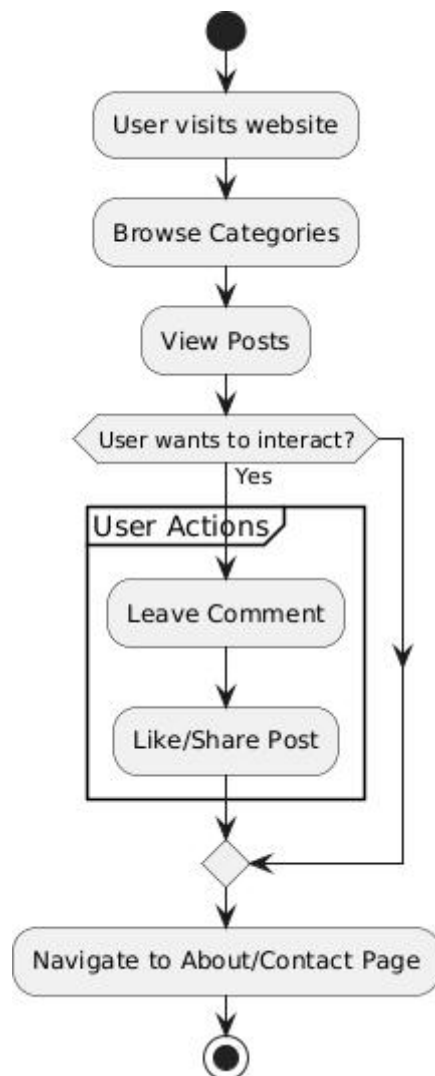
```
mysql>
```

### 4.3 PROCEDURAL DESIGN:

Process logic (flowchart ) of each module

#### **4.3.1 User Panel Design**

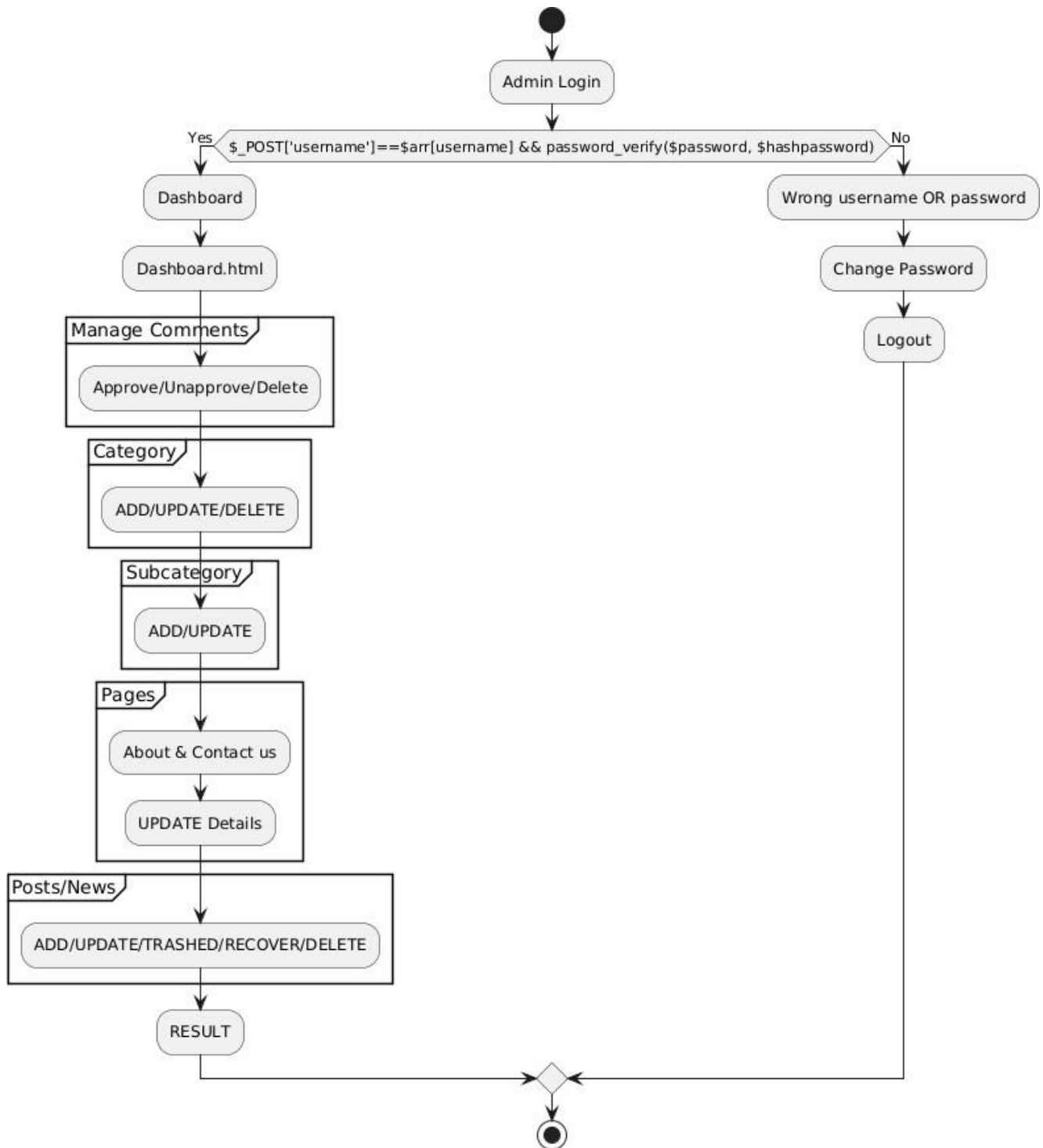
In user panel design we have done our task for user. Here we provide facility about Online News Paper. In index page user can select any options which is needed by him/her. By selecting options he/she can see the desired page. Then he/she can get the all oriented information finally. The design of user panel is shown in following flow chart....



**Fig. 4.1: The user panel flowchart part.**

### **4.3.2 Admin Panel Design**

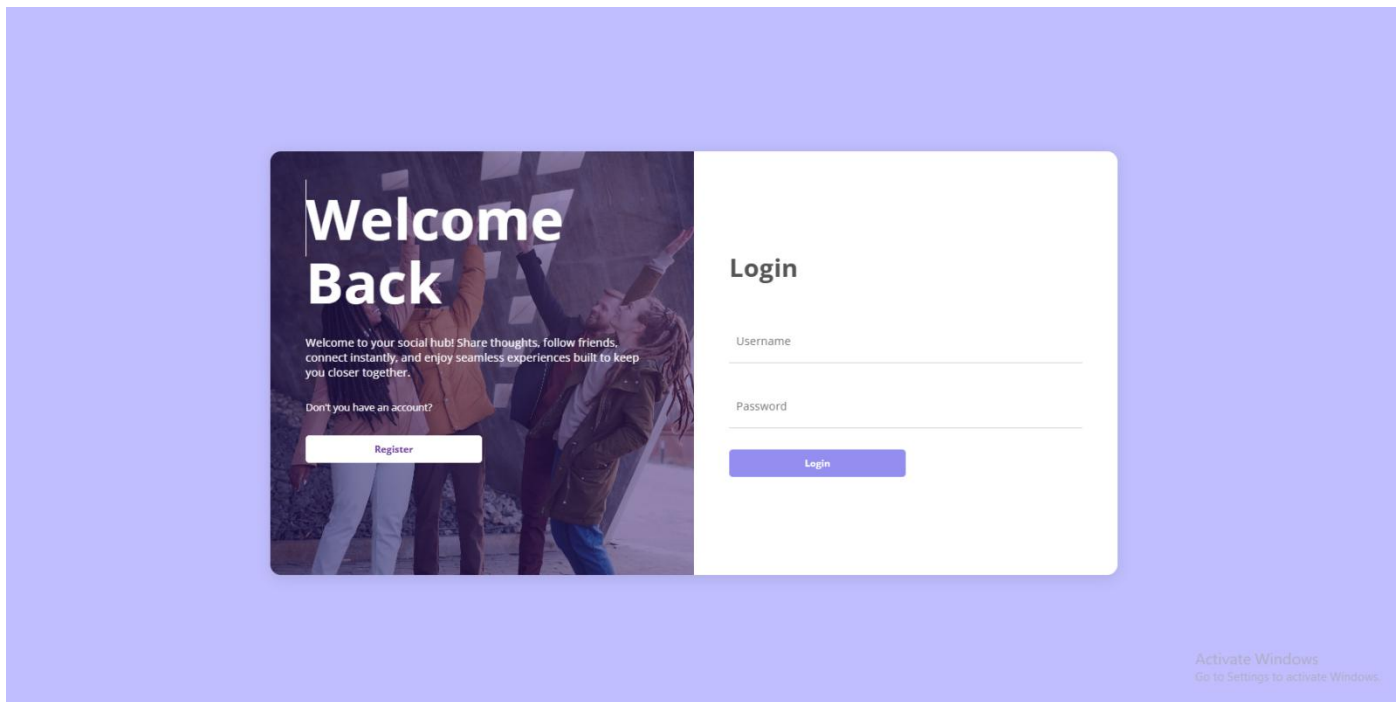
We have design user login facility to manage and update all of the information. It is fully secured page. Without appropriate username and password it cannot be accessed by anyone. For admin login after giving username and password we need to click a login button , when we click login button it is not directly entered in home page , it stay in login page. Then it starts a session and set two variables called username and password. If the username and password are matched with database, it can enter in home page. It is not possible without click login button. In case if username or password are not matched with database than Invalid username or password message is shown. We can describe the login facility in admin login by using below flow chart given below—



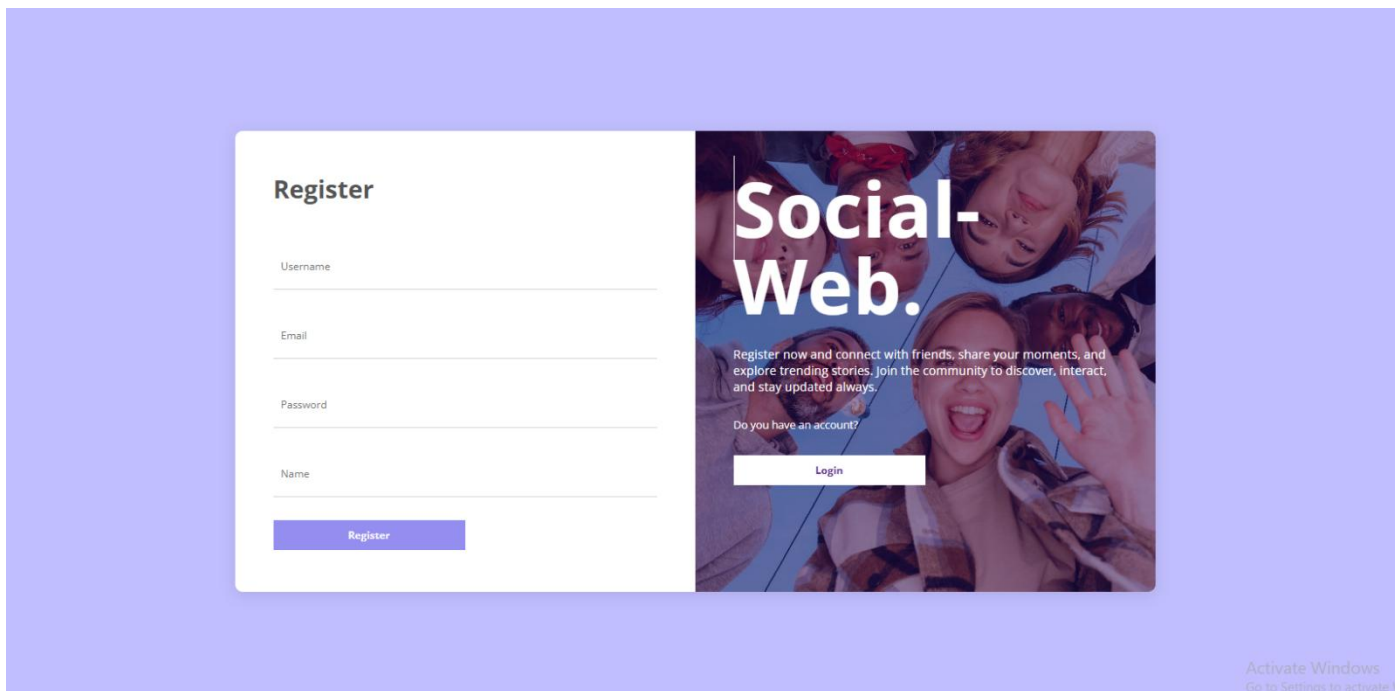
**Fig. 4.2: Admin Login Flow Chart.**

## SCREENSHOTS

### Login Page



### Register or Signup page :



## User Profile

Social-Web

Search...

Danish Shaikh

Friends

Groups

Marketplace

Watch

Memories

Your shortcuts

Events

Gaming

Gallery

Videos


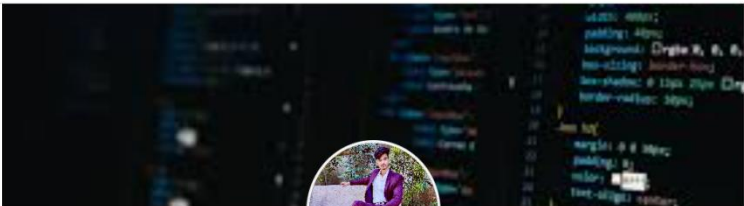
Messages

Others

Fundraiser

Tutorials

Courses




**Danish Shaikh**

Thane [www.Danishshaikh.in](http://www.Danishshaikh.in)











Update Logout

Danish Shaikh 2 hours ago




this is awesome



Suggestions For You

	Aarav Sharma	Follow	Dismiss
	Ananya Patel	Follow	Dismiss
	Rohan Gupta	Follow	Dismiss
	Sneha Reddy	Follow	Dismiss
	Vikram Singh	Follow	Dismiss
	Priya Sharma	Follow	Dismiss
	Aditya Verma	Follow	Dismiss
	Kavya Joshi	Follow	Dismiss
	Sameer Khan	Follow	Dismiss
	Isha Mehta	Follow	Dismiss

Latest Activities


	Aarav Sharma posted a new photo	2 min ago
	Ananya Patel updated her status	5 min ago
	Rohan Gupta liked a post	10 min ago

## Update Profile and Change Password :


Update Your Profile

Close

Cover Picture



Profile Picture



Email

Danish1@gmail.com

Password

123456

Name

Danish Shaikh

Country / City

Thane











Website

www.Danishshaikh.in




Update

Logout

Suggestions For You

	Aarav Sharma
	Ananya Patel
	Rohan Gupta
	Sneha Reddy
	Vikram Singh
	Priya Sharma
	Aditya Verma
	Kavya Joshi
	Sameer Khan
	Isha Mehta

Latest Activities

	Aarav Sharma
	Ananya Patel
	Rohan Gupta

**Home page**

Social-Web

Search...

Danish Shaikh

Friends

Groups

Marketplace

Watch

Memories

Your shortcuts

Events

Gaming

Gallery

Videos

Messaging

Others

Fundraiser

Tutorials

Courses

Danish Shaikh

+

Danish Shaikh

Rohan Sharma

Simran Patel

Arjun Verma

What's on your mind Danish Shaikh?

Add Image

Add Place

Tag Friends

Share

Tarvesh

an hour ago

Having fun!!!

Suggestions For You

Aarav Sharma

FollowDismiss

Ananya Patel

FollowDismiss

Rohan Gupta

FollowDismiss

Sneha Reddy

FollowDismiss

Vikram Singh

FollowDismiss

Priya Sharma

FollowDismiss

Aditya Verma

FollowDismiss

Kavya Joshi

FollowDismiss

Sameer Khan

FollowDismiss

Isha Mehta

FollowDismiss

Latest Activities

Aarav Sharma

Go to Settings to activate Windows.

posted a new photo

2 min ago



User Post:



Danish Shaikh

2 hours ago



hii every one



1 Likes

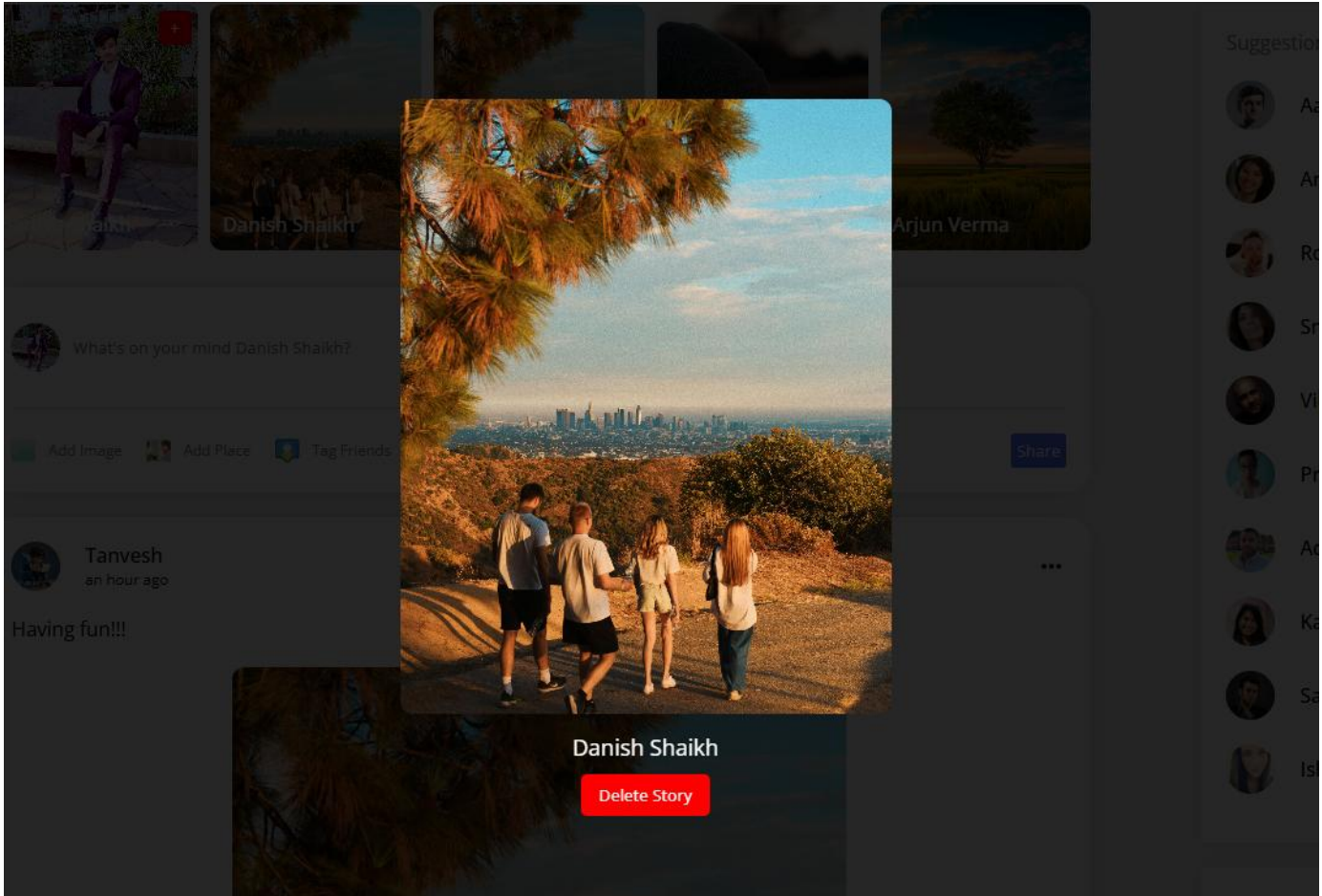


See Comments



Share

## User Story:



## Coding

### About.html

```
{% extends 'base1.html' %}

{% block content %}

{% load static %}

<div class="about-details section-padding30">

    <div class="container">

        <div class="row">

            <div class="offset-xl-1 col-lg-8">

                <div class="about-details-cap mb-50">

                    <h4>About Us</h4>

                    <p>{{ page.aboutus }} </p>

                </div> </div></div> </div> {% endblock %}
```

### Login.html

```
{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta content="width=device-width, initial-scale=1.0" name="viewport"

    <title>Online News Portal System|| Login Page</title>

    <!-- Favicons -->

    <link href="{% static 'assets/img/favicon.png'%}" rel="icon">

    <link href="{% static 'assets/img/apple-touch-icon.png'%}" rel="apple-touch-icon">

    <!-- Google Fonts -->

    <link href="https://fonts.gstatic.com" rel="preconnect">
```

```
<link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Nunito
:300,300i,400,400i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i"
rel="stylesheet">
```

```
<!-- Vendor CSS Files -->
```

```
<link href="{% static 'assets/vendor/bootstrap/css/bootstrap.min.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/bootstrap-icons/bootstrap-icons.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/boxicons/css/boxicons.min.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/quill/quill.snow.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/quill/quill.bubble.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/remixicon/remixicon.css'%}" rel="stylesheet">
```

```
<link href="{% static 'assets/vendor/simple-datatables/style.css'%}" rel="stylesheet">
```

```
<!-- Template Main CSS File -->
```

```
<link href="{% static 'assets/css/style.css'%}" rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
<main>
```

```
<div class="container">
```

```
<section class="section register min-vh-100 d-flex flex-column align-items-center justify-content-
center py-4">
```

```
<div class="container">
```

```
<div class="row justify-content-center">
```

```
<div class="col-lg-4 col-md-6 d-flex flex-column align-items-center justify-content-center">
```

```
<div class="d-flex justify-content-center py-4">
```

```
<a href="{% url 'index' %}" class="logo d-flex align-items-center w-auto">
```

```
<span class="d-none d-lg-block">Online News Portal</span>
```

```
</a>
```

```
</div><!-- End Logo -->
```

```
<div class="card mb-3">
```

```
<div class="card-body">
```

```

<div class="pt-4 pb-2">
  <h5 class="card-title text-center pb-0 fs-4">Login to Your Account</h5>
  <p class="text-center small">Enter your username & password to login</p>
</div>

{% if messages %}
{% for message in messages %}
  {% if message.tags == 'error' %}
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  {{ message }}
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
  </button>
</div>
{% endif %}
{% endfor %}
{% endif %}

  {% if messages %}
  {% for message in messages %}
    {% if message.tags == 'success' %}
<div class="alert alert-warning alert-dismissible fade show" role="alert">
  {{ message }}
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
<span aria-hidden="true">&times;</span>
  </button>
  </div>
{% endif %}
{% endfor %}
{% endif %}

```

```
<form class="row g-3 needs-validation" method="post" action="{% url 'doLogin' %}">
    {% csrf_token %}
    <div class="col-12">
        <input type="hidden" name="next" value="{{ next }}">
        <label for="yourUsername" class="form-label">Username</label>
        <div class="input-group has-validation">
            <span class="input-group-text" id="inputGroupPrepend">@</span>
            <input type="text" name="username" class="form-control" id="username" required
value="{{ username|default:'' }}">
            <div class="invalid-feedback">Please enter your username.</div>
        </div>
    </div>
    <div class="col-12">
        <label for="yourPassword" class="form-label">Password</label>
        <input type="password" name="password" class="form-control" id="password" required>
        <div class="invalid-feedback">Please enter your password!</div>
    </div>
    <div class="col-12">
        <button class="btn btn-primary w-100" type="submit">Login</button>
    </div>
    <div class="col-12">
        <p class="small mb-0" style="font-size: large;"><a href="{% url 'index' %}">Home</a></p>
    </div>
</form>
</div>
</div> </div> </div> </section> </div> </main> <!-- End #main -->
```

```

<a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

<!-- Vendor JS Files -->
<script src="{% static 'assets/vendor/apexcharts/apexcharts.min.js'%}"></script>
<script src="{% static 'assets/vendor/bootstrap/js/bootstrap.bundle.min.js'%}"></script>
<script src="{% static 'assets/vendor/chart.js/chart.umd.js'%}"></script>
<script src="{% static 'assets/vendor/echarts/echarts.min.js'%}"></script>
<script src="{% static 'assets/vendor/quill/quill.js'%}"></script>
<script src="{% static 'assets/vendor/simple-datatables/simple-datatables.js'%}"></script>
<script src="{% static 'assets/vendor/tinymce/tinymce.min.js'%}"></script>
<script src="{% static 'assets/vendor/php-email-form/validate.js'%}"></script>

<!-- Template Main JS File -->
<script src="{% static 'assets/js/main.js'%}"></script>

</body></html>

```

### [Profile.html](#)

```

{% extends 'base.html' %}
{% block content %}
<div class="pagetitle">
  <h1>Profile</h1>
  <nav>
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="{% url 'dashboard' %}">Home</a></li>
      <li class="breadcrumb-item">Profile</li>
    </ol></nav>
</div><!-- End Page Title -->
<section class="section">
  <div class="row">
    <div class="col-lg-12" style="padding-left: 100px;padding-right: 100px;">

```

```

<div class="card">
  <div class="card-body">
    <h5 class="card-title">Profile Details</h5>
    {% if messages %}
    {% for message in messages %}
      {% if message.tags == 'error' %}
        <div class="alert alert-warning alert-dismissible fade show" role="alert">
          {{message}}
          <button type="button" class="close" data-dismiss="alert" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button></div>
        {% endif %} {% endfor %} {% endif %}
      {% if messages %}
      {% for message in messages %}
        {% if message.tags == 'success' %}
          <div class="alert alert-warning alert-dismissible fade show" role="alert">
            {{message}}
            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
              <span aria-hidden="true">&times;</span>
            </button></div>
          {% endif %} {% endfor %} {% endif %}
        <!-- Vertical Form -->
        <form class="row g-3" method="POST" action="{% url 'admin_profile_update' %}"
        enctype="multipart/form-data">
          {% csrf_token %}
          <div class="col-12">
            <label for="inputNanme4" class="form-label">Old Profile Pic</label>
            
          </div>

```



```

<div class="col-12">
    <label for="inputNanme4" class="form-label">Profile Pic</label>
    <input type="file" class="form-control" name="profile_pic">
</div>
<div class="col-12">
    <label for="inputEmail4" class="form-label">First Name</label>
    <input type="text" class="form-control" name="first_name" value="{{user.first_name}}">
</div>
<div class="col-12">
    <label for="inputPassword4" class="form-label">Last Name</label>
    <input type="text" class="form-control" name="last_name" value="{{user.last_name}}">
</div>
<div class="col-12">
    <label for="inputAddress" class="form-label">Email</label>
    <input type="email" class="form-control" readonly="True" name="email"
value="{{user.email}}">
</div>
<div class="col-12">
    <label for="inputAddress" class="form-label">Username</label>
    <input type="text" class="form-control" readonly="True" name="username"
value="{{user.username}}">
</div>
<div class="text-center">
    <button type="submit" class="btn btn-primary">Submit</button>
    <button type="reset" class="btn btn-secondary">Reset</button>
</div>
</form><!-- Vertical Form -->
</div>
</div></div> </section>{% endblock %}

```

## Testing

**Unit Testing:** Unit testing where individual program units or object classes are tested. Here by using this testing we have focused on testing the functionality of methods.

**Module Testing:** Where this is the combination of unit program is called module. Here we tested the unit program (5-6 programs) is where the module programs have dependency.

**Sub-system Testing:** Then we combined some module for the Preliminary System Testing in our Project.

**System Testing:** Where it is the combination of two or more sub-system and then it is tested. Here we tested the Entire system as per the requirements.

**Acceptance Testing:** Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting this project to User then they tested it and to determine whether to accept application. It is the system testing performed by the customer(s) to determine whether they should accept the delivery of the system.

## **6. Future scope**

- The Online News Portal can be enhanced with the following features:
- The Social Media Website can be enhanced with the following features:
- AI-based Personalized Feeds for better user engagement
- Mobile Application for Android & iOS platforms
- Real-time Messaging & Notifications using WebSockets
- Video & Live Streaming Support
- Advanced Search with Filters (posts, users, hashtags, locations)
- Multi-language Support to connect global audiences
- AI-driven Content Moderation & Spam Detection
- Marketplace & In-App Purchases for business opportunities
- Analytics Dashboard for users and admins to track engagement
- Integration with third-party platforms (e.g., Instagram, LinkedIn, Twitter)

## **Conclusion**

The Social Media Website Project provides a modern, interactive, and scalable platform for users to connect, share, and engage with one another. It ensures a seamless user experience with features like posting, liking, commenting, and real-time interactions. The system leverages technologies (MySQL, Express.js, React, Node.js) to deliver a secure and efficient platform. Role-based access for users and admins ensures safety, while cloud-based storage and authentication guarantee reliability. Designed with scalability and future expansion in mind, the platform is well-equipped to integrate advanced AI-driven features, making it a robust and competitive solution in the evolving digital landscape.

## BIBLIOGRAPHY

Full-Stack React Projects by Shama Hoque

Learning React: Modern Patterns for Developing React Apps by Alex Banks & Eve Porcello

MongoDB: The Definitive Guide by Kristina Chodorow

Node.js Design Patterns by Mario Casciaro & Luciano Mammino

---

## REFERENCES

**React Official Documentation** – <https://react.dev/>

**Node.js Official Documentation** – <https://nodejs.org/en/docs/>

**Express.js Guide** – <https://expressjs.com/>

**MongoDB Official Docs** – <https://www.mongodb.com/docs/>

**JWT (JSON Web Token) Documentation** – <https://jwt.io/introduction>

**REST API Best Practices** – <https://restfulapi.net/>

**Axios (HTTP Client)** – <https://axios-http.com/>

**GitHub Docs** – <https://docs.github.com/>

**Visual Studio Code** – <https://code.visualstudio.com/>

**Postman API Platform** – <https://www.postman.com/>