PROJECT SYNOPSIS ON

Tweetzy: A Social Media Hub SUBMITTED

TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING FOR

Full Stack Engineering(22CS037)

Submitted By:

Name(s): Kavya, Khushi Garg

University Roll No(s).: 2210991776, 2210991800 Semester: 6’th

Session: 2024-25

Index

Sr. no Topic Page No

1. [Problem Statement 3](#_bookmark0)
2. [Title of project 3](#_bookmark1)
3. [Objective & Key Learning’s 3](#_bookmark2)
4. [Options available to execute the project 3-4](#_bookmark3)
5. [Advantages/ Disadvantages 4](#_bookmark4)
6. [References 5](#_bookmark5)

### Problem Statement:

Lack of Time for Meaningful Connections: People struggle to foster genuine relationships due to busy schedules and fragmented

social interactions. • Mental Health Strain: Many users face stress and anxiety from the pressures of modern life, with social media often exacerbating these

mental health challenges. • Limited Local Information: Users often have limited access to relevant content and events specific to their local area, reducing the

platform’s usefulness. • Complicated User Interface: Cluttered and complex app designs create a frustrating experience, making it difficult for users to navigate

and engage

### Title of project:

Tweetzy: A Social Media Hub.

### Objective & Key Learnings:

The objective of Tweetzy is to transform the social media landscape by:

Encouraging authentic and meaningful interactions over superficial engagement.

Promoting mental well-being through digital detox, mindfulness reminders, and screen time tracking. Delivering personalized, location-based content for enhanced user relevance.

Ensuring a minimalist and user-friendly interface for seamless navigation and engagement.

Key Learnings:

* + User Authentication & Authorization: Implementing login, signup, and session management securely
  + Real-time Features: Implementing WebSockets for live notifications, tweet updates, and user interactions.
  + Scalability & Performance : Optimizing database queries, implementing caching, and using pagination.
  + Security Best Practices: Preventing SQL injection, XSS, CSRF attacks, and securing user data.
  + Backend Development: Using Node.js, Express.js, to manage logic and data flow.
  + Frontend Development: Building an interactive UI with React.js.

## Options available to execute the project:

* + Web-Based Platform (MERN stack + React)
    - Suitable for desktop and mobile users.
    - Easier to integrate with cloud-based analytics and tracking.
  + Mobile Application ( React Native)
    - Offers better accessibility and push notifications for real-time updates.
  + Cloud-Based Solution (AWS, Firebase, or Google Cloud)
    - Enables seamless data synchronization and scalability.
    - Provides security, storage, and backup management.

## Advantages/ Disadvantages:

Advantages:

* + **Hands-on Experience** – Helps in learning full-stack development (frontend, backend, database, APIs).
  + **Real-world Application** – Mimics a production-level social media platform, preparing for real- world projects.
  + **Scalability & Performance Optimization** – Teaches how to handle large user interactions efficiently.
  + **Security Practices** – Provides experience in handling authentication, authorization, and data security.
  + **Understanding Social Media Mechanics –** Helps in learning features like likes, retweets, comments, and real-time updates
  + **Customization & Creativity** – Can modify and add unique features like AI-based recommendations or advanced analytics.

Disadvantages:

* + **Time-Consuming** – Developing a fully functional clone requires significant time and effort.
  + **Complexity** – Handling real-time updates, notifications, and large-scale data management can be challenging.
  + **Scalability Issues** – Without proper optimization, the system may slow down with increased user activity.
  + **Resource Requirement** – Deploying a production-level version may require advanced hosting solutions, increasing costs.
  + **Security Risks** – If not properly secured, it can be vulnerable to cyber threats like SQL injection, CSRF, or XSS.

# REFERENCES

* + **Node.js:** Official Documentation
  + **Express.js:** Documentation
  + **MongoDB:** Basics
  + **EJS:** Documentation
  + **GitHub Actions:** Documentation
  + **React:** Documentation