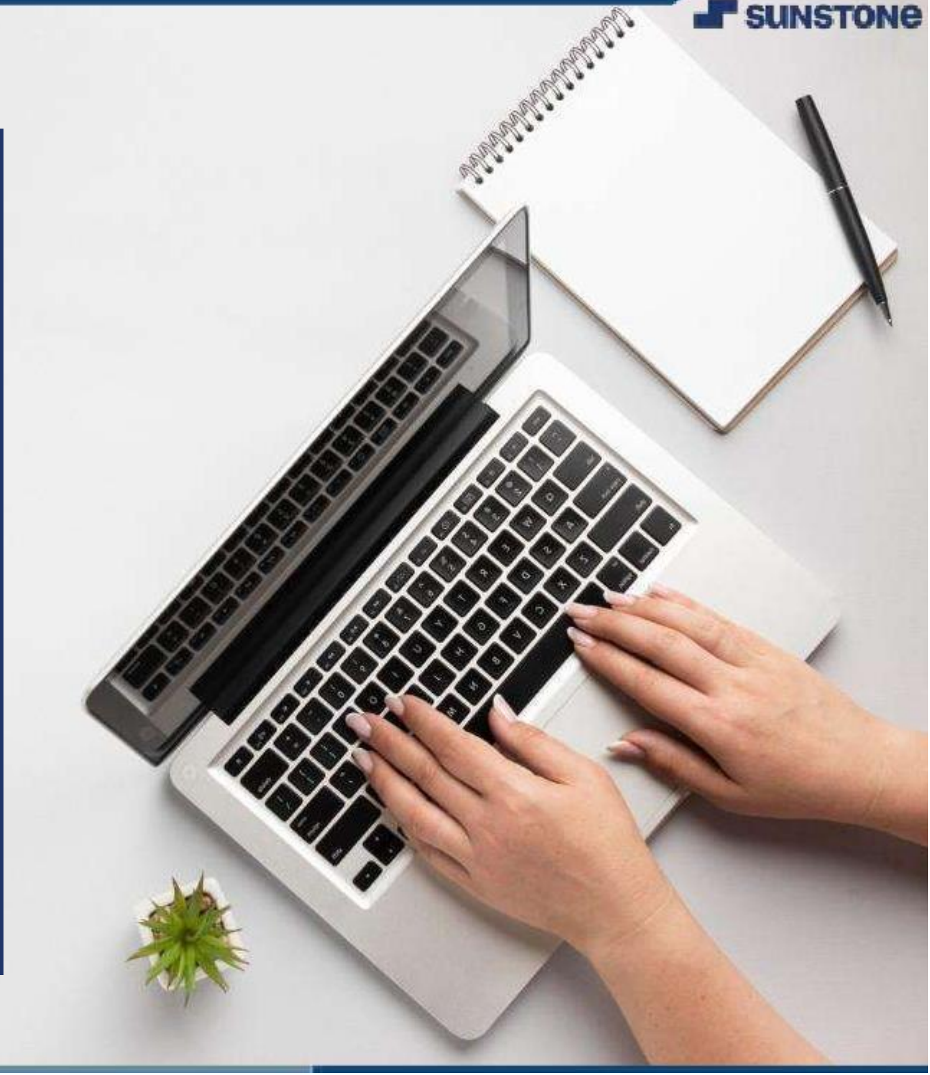
SUNSTONE

“Social Media Website”

Task - 4

Social Media Website

A social media website is an online platform that enables users to create profiles, share content, and connect with others. Users can post text, photos, videos, and links, and interact with each other through comments, likes, and shares. Popular features include timelines, news feeds, and messaging. Social media sites often cater to specific interests or demographics, fostering communities around shared topics. Privacy settings allow users to control the visibility of their content. Monetization may occur through ads, sponsored content, or premium features. Key examples include Facebook, Instagram, Twitter, and LinkedIn.



Task 4 :: Dynamic Frontend

Integrate the APIs to frontend to ensure the dynamic feature of website

- Point base api to the severs base url
- Design api calls for each element
- Handle errors in the output
- Render output of apis to different low level components
- Secure content of post apis

Evaluation Metric:

100% Completion of the above tasks

Learning outcome

- Developing complicated UI using HTML components
- Using props drilling and context to pass variables
- Getting familiar with different type of api calls
- Handling different input data

Step-Wise Description

Step 1: **Identify Platform Requirements**: Determine the social media platforms your app will integrate with (e.g., Facebook, Twitter, Instagram).

Step 2:**Review API Documentation**: visit the official developer documentation for each social media platform to understand their APIs.

Step3 :**Authentication Setup**:learn about the authentication methods required by each API (Oauth, API keys) and implement the necessary authentication process.

Step4:**Select Relevant Endpoints** : Identify the specific API endpoints that meet your app's needs (e.g., user data, posts, comments).

Step 5:**Understand Rate Limits** : Be aware of any rate limits imposed by the social media APIs to ensure your app stays within acceptable usage limits.

Step6:**Data Handling and Formatting** : Understand the data format (JSON, XML) returned by the API and implement proper handling and formatting in your app.

Step 7:**Error Handling**: Implement error handling mechanisms to gracefully manage issues that may arise during API requests.

Step-Wise Description

Step 8: Set Up Your Front-End Project:

Ensure your front-end project is well-organized with separate folders for HTML, CSS, and JavaScript.

Create a folder specifically for handling API-related functions**.**

Step 9: Testing: Create a testing environment to simulate API requests and responses, ensuring your app interacts correctly with the social media APIs.

Step 10 :User Permissions and Privacy: Familiarize yourself with the permissions model of each social media platform and ensure your app complies with user privacy guidelines.

Step 11:Stay Updated: Regularly check for updates or changes in the API documentation to adapt your app accordingly.

step 12:Monitor Usage and Analytics : Implement analytics to monitor API usage and user interactions within your app**.**

Summary of your task

This Software Requirements Specification (SRS) outlines the requirements for a portfolio website developed with HTML, CSS, and JavaScript.
SRS in social media apps encompasses features related to connecting, communicating, and engaging with other users. It may include components such as friend requests, messaging systems, follower/following mechanisms, and algorithms for content discovery based on social connections. The goal is to enhance the social experience within the app by providing tools and services that support and strengthen relationships among users.

Tech Stack: React. Js for the frontend, Node. Js with Express for the backend, and MongoDB
Bootstrap 4 and Redux for the database.
Functionality: include user authentication, profile creation, posting content, following/friending other users, liking or commenting on posts, notifications, and a newsfeed. Additionally, features like direct messaging, search functionality, and privacy settings are often essential.
Tools: npm for package management, Axios for API requests, React Router for navigation, and Mongoose for MongoDB connectivity.

Submission Github

https://github.com/Prema123-sm/naan-mudhalvan

Thank you!