Project Title	Survey Application
Technologies	MERN
Domain	Utility
Project Level	Medium

Table

Table of Contents

3. Submission requirements:

1. Problem Statement:

Design a MERN stack-based Survey Application

1.1. Overview of MERN Stack and Survey App

Full-stack web applications can be launched more quickly and easily with the JavaScript stack MERN. The MERN Stack is comprised of the four technologies MongoDB, Express, React, and Node.js. The goal is to simplify and streamline the development process.

Each of these four potent technologies provides a complete environment for developers to operate in and makes a substantial contribution to the creation of web apps.

A website or application where users may sign up and post surveys so that others can join and respond to those surveys. Survey software is quite popular since it allows us to understand the user's needs and wants in order to produce more effective products and services.

Note: To construct a user-friendly and appealing survey app, you can look to a variety of survey web applications.

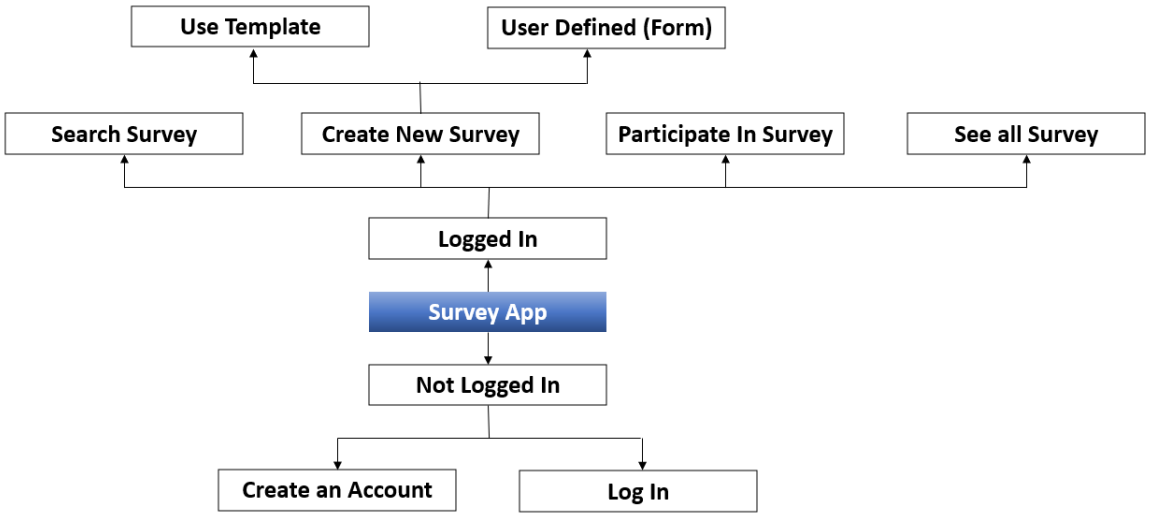
1.1.1 The user's survey app should provide at least these features and functions.

These are the characteristics of your survey application that we can anticipate seeing-

a. When visiting the website for the first time or logging in to his account, the user should have the choice to register or log in (if already registered).
b. He can design new surveys after signing into his account and can also take part in others' surveys to share his thoughts on those.
c. When designing a survey, a user should have access to at least 10 pre-defined templates, such as MCQ, descriptive, optional, poll template, etc.

d. To prevent fraudulent submissions, only users who have registered on the website are able to view or take part in surveys.
e. Toggling between dark and light modes should be an option for the user to achieve the best viewing experience.
f. The user can adjust the templates accordingly.
g. The webpage ought to adapt to all screen sizes.
h. The user specifies the survey's category when constructing it, such as educational, marketing, movies, etc. The user must choose from among these predefined categories in the application.
i. The user can narrow down the surveys by category to only see the surveys that interest him.
j. The survey title or category, which will be provided to every survey at the time of creation, should be able to be used by a user to search a survey.

Low Level user flow



1.2. Project Objective

• **Survey App:** Everyone has an opinion on every subject in the present era, and people love to voice them. This website will be very beneficial in helping you do that. A user can quickly design a new poll and take part in others to share his opinions on that.

• **Simple and User Friendly:** The website will offer a user-friendly user interface so that anyone can utilize it with ease.

1.3. Scope of The Project

1. The ideal location for a business to learn about the preferences of its clients is through survey apps. He can behave in a way that suits their preferences in order to increase leads and

sales. If we are aware of the needs of the user, we can meet those needs and innovate in order to give him a positive experience.

2. People will have free access to all of these amenities.

1.4. Functional and Non-Functional Requirements: -

1.4.1. Functional Requirements

1. **User Registration:** An email address, a username, and a password must be required for the user to register for the application. Users must be able to register themselves upon opening the application, or if they already have an account, they can log in immediately.

2. **Adding New Surveys:** New surveys can be added to the application by the user.

3. **Participation:** Any user who is logged in may take part in any of the current surveys.

5. **Delete Survey:** The survey's creator has the right to delete it whenever he wants.

6. **Dashboard**: The user's dashboard displays all of the surveys he or she has created along with the results. Any survey created by the user cannot be changed, his only option is to remove it.

1.4.2. Non-Functional Requirements

1. **Privacy:** His profile and surveys are only visible to logged-in users. Only the user's own surveys can be deleted.

2. **Robustness:** The user data should be kept on a server in case the user's machine crashes so that they can access it once the system has been repaired.

3. **Performance:** The application must be lightweight in order to load quickly.

1.5. Use Case Table

Authentication System	Register, Login, Logout	User
Search Box	Search Surveys	User
User Dashboard	Display all the created surveys and profile data	User

Table 1. Use Case

2. Project Evaluation Metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).

- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Implementation of reverse proxy, load balancing, and security group is mandatory for deployed applications.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Low-level Document (LLD), and Wireframe documents**.**

2.6. System Architecture:

You have to submit a system architecture design in your wireframe document and architecture document.

2.7. Optimization of solutions:

Try to optimize your solution on code level, architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:

You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.