

Survey App

Assignment Category: assignment12_category_0020

A company is seeking a talented Developer to contribute to developing a Survey application using the MERN (MongoDB, Express.js, React.js, Node.js) stack. This project involves integrating payment functionalities, implementing a robust user management system, and creating an admin dashboard with role management. As a vital team member, you will be pivotal in crafting a feature-rich platform for survey creation, voting, result analysis, and user interaction.

What is a survey?

A survey gathers information or feedback from individuals to understand their opinions, preferences, or experiences on a particular topic or subject. Surveys can be conducted through various mediums such as online forms, questionnaires, interviews, or telephone calls. They are widely used in market research, academic studies, customer satisfaction analysis, and organizational evaluations.

Here are example websites to get an idea:

- SurveyMonkey (www.surveymonkey.com)
- Google Forms (www.google.com/forms)
- Typeform (www.typeform.com)

🚩: 0 [If there are any updates, they will be mentioned here]

Key Rules

1. GitHub Commits:

- Minimum 20 notable commits on the client side.
- Minimum 12 notable commits on the server side.

2. README.md:

- Include website name, admin username, password, and live site URL.
- List at least 10 bullet points highlighting your website features.

3. Responsiveness:

- Ensure responsiveness for mobile, tablet, and desktop views, including the dashboard.

4. Private Route Handling:

- Users should not be redirected to the login page after reloading a private route.

5. Environment Variables:

- Use environment variables to hide Firebase config keys and MongoDB credentials.

6. Alerts and Notifications:

- Use sweet alerts/toasts/notifications for CRUD operations, successful login, and sign-up.
- Do not use default browser alerts.







7. Data Fetching:

- Implement TanStack Query for all GET method data fetching.

Main Features

Pages

1. Homepage

- **Hero Section** :
 - Banner with website overview and explore button.
 - Background image.
- **Featured Surveys Section** :
 - Display the 6 most voted surveys from the database.
- **Latest Surveys Section** :
 - Display 6 most recently created surveys from the database.
- **How It Works Section** :
 - Content matching website theme.
- **FAQ**  :
 - Meaningful FAQs about the website and features.

2. Surveys Page (Public):

- Display all surveys with title, short description, and total votes.
- Filters for the category and
- Sort for vote count.

3. Survey Details Page (Public):

- Display detailed survey information.
- Allow voting for logged-in users only.
 - It would have a submit button. On clicking the submit button the survey will be submitted.
 - Show these screenshots [image-1](#), [image-2](#), [image-3](#), [image-4](#)
 - This page will be public. Unauthenticated users won't be able to vote.
- Only pro-users can add comments.
- Display survey results visually after voting or after the survey deadline.
- Allow users to report inappropriate surveys.

4. Pricing Page (Public):

- Integrate payment system for pro-user membership.
- Pro-user nav link for navigation to the payment page.
- Change user role to **pro-user** upon successful payment.

User Authentication

- Email/password account creation.
- Social media authentication.

Role Management

- **User roles:** user, surveyor, admin, pro-user.
- The default role for new users is user.

Dashboard

1. Surveyor Dashboard: (/dashboard/surveyor)

- Survey creation with questions. (/dashboard/surveyor/create)
- A question will contain the following information.
 - Title.
 - Description.
 - Options. (yes or no).
 - You can make two fields yes or no. then you can count the value.

- Category.
 - Select from a dropdown
 - You can search for the most common survey category from Google or chatGPT.
- Deadline. (type date)
- The following items should not be in the form and add this from the backend:
 - Status (publish/unpublish).
 - default status: publish.
 - Timestamp (Survey creation time). (type date)
- Show these screenshots [image-1](#), [image-2](#), [image-3](#), [image-4](#)
- Survey update. (/dashboard/surveyor/update/:id)
 - Form fields: Title, Description, Options. (yes or no), Category, Deadline.
- View individual survey responses for individual surveys added by the currently logged-in user in tabular form with a details button. (/dashboard/surveyor/surveys)
- Upon clicking the details button, show individual survey responses with detailed information and users. (/dashboard/surveyor/surveys/:id)

Serial No	User Email	User Name	Vote
-----------	------------	-----------	------

- **Note:** only the surveyor can create a survey.

2. Admin Dashboard: (/dashboard/admin)

- Manage users and roles. (/dashboard/admin/users)
 - Filter users by role. (/dashboard/admin/users)
- Publish/unpublish surveys. (/dashboard/admin/surveys)
- View all payments and survey responses. (/dashboard/admin/payments)

3. User Dashboard: (/dashboard/user)

- Participate in surveys and (/dashboard/user/surveys)
- Reported surveys. (/dashboard/user/my-reports)
- a. **Pro-User: (only pro-user can access this page)** (/dashboard/user/comments)
 - Commented on surveys.

Challenge Requirements 🌟

- The feedback message is required when an admin is unpublishing a survey.
 - A feedback modal will be open on unpublishing a survey with a feedback input field and a button. This feedback will be shown on the (/dashboard/surveyor/feedback) page.
- View survey feedback added by the admin on unpublish a survey in tabular form. (/dashboard/surveyor/feedbacks)
- Toggle survey responses between table and chart views. (/dashboard/surveyor/surveys/:id)
 - You can use any bar chart, pie chart, etc. (show chart based on **yes**, **no** vote count)
- Surveys may contain multiple questions. See [image](#).
- JWT Authentication 🔑
 - Create and store JWT tokens in local storage for authentication.
 - Implement JWT on private routes.
 - Optional 401 and 403 handling.

Optional Features

- Animation on the homepage sections.
- Toggle system between card and table views.
- Implement like and dislike features for individual surveys.

Optional Package Requirements: (Use any 3 of the following:)

- Headless UI / shadcn/ui
- Yup (for form validation)
- Mongoose
- Moment

Extra Pages Recommendations 📄

- About Us
- Contact Us
- Privacy Policy and Terms of Service
- Help Center

Guidelines

- Spend 15-20 minutes on core features.
- Start with a basic idea and progressively add features.
- Prioritize user experience, data security, and seamless payment integration.
- Use ChatGPT for generating sample data initially.
- Regularly commit and push the code.
- Screenshots [link](#).

Submission Requirements:

- **Assignment Category:**
- **Admin Email:**
- **Admin Password:**
- **Front-end Live Site Link:**
- **Client Side GitHub Repository Link:**
- **Server-Side GitHub Repository Link:**