1.11

File Verification Application

Service Monitoring Application

Survey Application

The task will involve creating a simple Survey application. The application will allow users to respond to survey questions and view a list of all responses to the survey questions.

The task will be split into 3 components:

1. Databases

Using a Relational Database Management System, either MySQL, Oracle, Microsoft SQL Server or Postgres:

- 1. Design an Entity Relationship Diagram (ERD) for the database for the application.
- 2. Implement the database, name it sky_survey_db.

2. REST API

Using your preferred language, create a REST API that connects to your database.

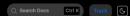
The API should have the following endpoints:

- 1. To fetch list of questions
- url: /api/questions
- · method: GET
- response:

- 2. To submit responses to the questions
- UT: /api/questions/responses
- method: PUT
- response:

The API should support uploading of files through use of form-data.

3. To fetch submitted responses to the questions



ON THIS PAGE

- 2 Hear Interfere (Mahile or Web)
- 1. Survey Form
- 2. Survey Responses
- 2.1 Requirements
- Task guidelines

- Url: /api/questions/responses
- · method: GET
- response:

The API should support:

- · pagination of the records.
- filtering of the responses based on email_address
- 4. To download a certificate by providing the id of the certificate as a URL Parameter
- UTI: /api/questions/responses/certificates/{id}
- · method: GET

Provide a **Postman Collection** documenting the endpoints above with their saved responses

3. User Interface (Mobile or Web)

Create a User Interface for the application.

- For mobile developers, use your preferred mobile development languages or framework i.e. Android, Flutter or React Native.
- For web & backend developers, use your preferred web development languages or framework.

The User Interface should have two pages:

- 1. Survey Form
- 2. Survey Responses

1. Survey Form

The page will have the form through which users can respond to the questions.

1.1 Requirements

- 1. The form should be a stepped form; with question as a step.
- $2. \ The \ list of \ questions \ should \ be \ fetched \ by \ making \ a \ request \ to \ the \ \textbf{Endpoint 1} \ in \ the \ \textbf{REST API} \ section \ above$
- 3. The form should have **Next** and **Previous** button to navigate through each question.
- 4. On the first question, the Previous button should be hidden
- 5. For questions with required yes, ensure the user provides a response before proceeding to the next question
- 6. The final step should have a preview of all the collected data and a **Submit** button to submit the collected data.
- 7. On clicking the Submit button, the responses should be submitted to the database via the Endpoint 2 in the REST API section above
- 8. Use the appropriate form input for each question i.e. $long_text \Rightarrow textarea$

2. Survey Responses

The page will be used to show the submitted responses to the questions.

2.1 Requirements

- 1. Fetch the list of submitted responses using **Endpoint 3** in the **REST API** section above
- 2. The list should be paginated
- 3. You should be able to filter the responses using the $\mbox{\bf email_address}$
- 4. You should be able to download the certificates using the Endpoint 4 in the REST API section above

Add a navigation to be able to switch between the two pages.

Task guidelines

You are expected to do all these components of the application:

- 1. Database
- 2. REST API
- 3. User Interface i.e. either Web or Mobile

Simple survey client to store your:

1. ERD Diagram

2. database SOL file

3. REST API code

4. Postman Collection, each repository should have a REAGE, and documenting:

• how to set up and run your application on a local machine

• the deployment process (OPTIONAL but more points to those who can).

2. Deploy your application and provide a public URL to access it. For the mobile developers, generate an APK of your application and have it on the simple survey-client GitHub Repository. (OPTIONAL but more points to those who can)

3. Timeline to complete the project is 2 weeks. You will be expected to submit the following:

1. Link to the simple survey-client GitHub Repository

2. Link to the simple survey-client GitHub Repository

3. Link to the simple survey-client GitHub Repository

3. Link to the simple survey- spi GitHub Repository

3. Link to the public deployed web application for web & backend developers. (OPTIONAL but more points to those who can).

• Ensure the above attached are fully functional.

1. Create **public** GitHub repositories to version your source code:

Next pag
File Verification Application