# Online Survey G Feedback System

## Submitted by:

Name : Johnson Kumar

U.I.D : 23bcs12654

## Project Description

The *Online Survey & Feedback System* is a web-based application designed to create, manage, and analyse surveys efficiently. It enables administrators to build customized surveys, participants to provide responses through a user-friendly portal, and stakeholders to evaluate results using interactive dashboards. Built using *Spring Boot* for backend services, *React.js* for the frontend interface, and *MongoDB* for scalable data storage, the system provides a dynamic and secure platform for conducting surveys across academic, corporate, and research domains.

### Key Features:

- Dynamic survey creation with multiple question types.
- Secure participant portal for submitting responses.
- Real-time aggregation and visualization of results.
- Role-based access for administrators, creators, and participants.

## Project Distribution into Modules

The project is divided into five core modules to ensure clarity, scalability, and effective development:

### 1. Frontend – Survey Builder & Participant Portal (React.js)

- Survey Builder UI:
  - Create dynamic forms (MCQs, text fields, ratings, etc.).
  - Add, edit, and arrange questions.
  - Preview surveys before publishing.
- Participant Portal UI:
  - Display available surveys.

    - Provide responsive forms to submit answers.
    - Ensure a clean and mobile-friendly design.

## 2. Frontend – Result Dashboard & Admin Panel (React.js)

- Result Dashboard:
    - Show aggregated results in visual charts/graphs (Chart.js/Recharts).
    - Enable filtering by date, survey type, or user group.
    - Provide export options (CSV, Excel, PDF).
- Admin Panel:
    - Manage surveys (publish/unpublish/delete).
    - Control user roles (Admin, Creator, Participant).
    - Monitor participation rates and activity.

## 3. Backend & Database (Spring Boot + MongoDB)

- API Development (Spring Boot):
    - CRUD APIs for surveys and responses.
    - Authentication APIs (login, signup).
    - Secure communication with JWT tokens.
- Database (MongoDB):
    - Store surveys in JSON format (flexible schema).
    - Store participant responses and link them to surveys.
    - Optimize queries for fast analytics.
- Integration:
    - Connect backend APIs with frontend React components.
    - Deploy backend and database on cloud platforms (Render, MongoDB Atlas).

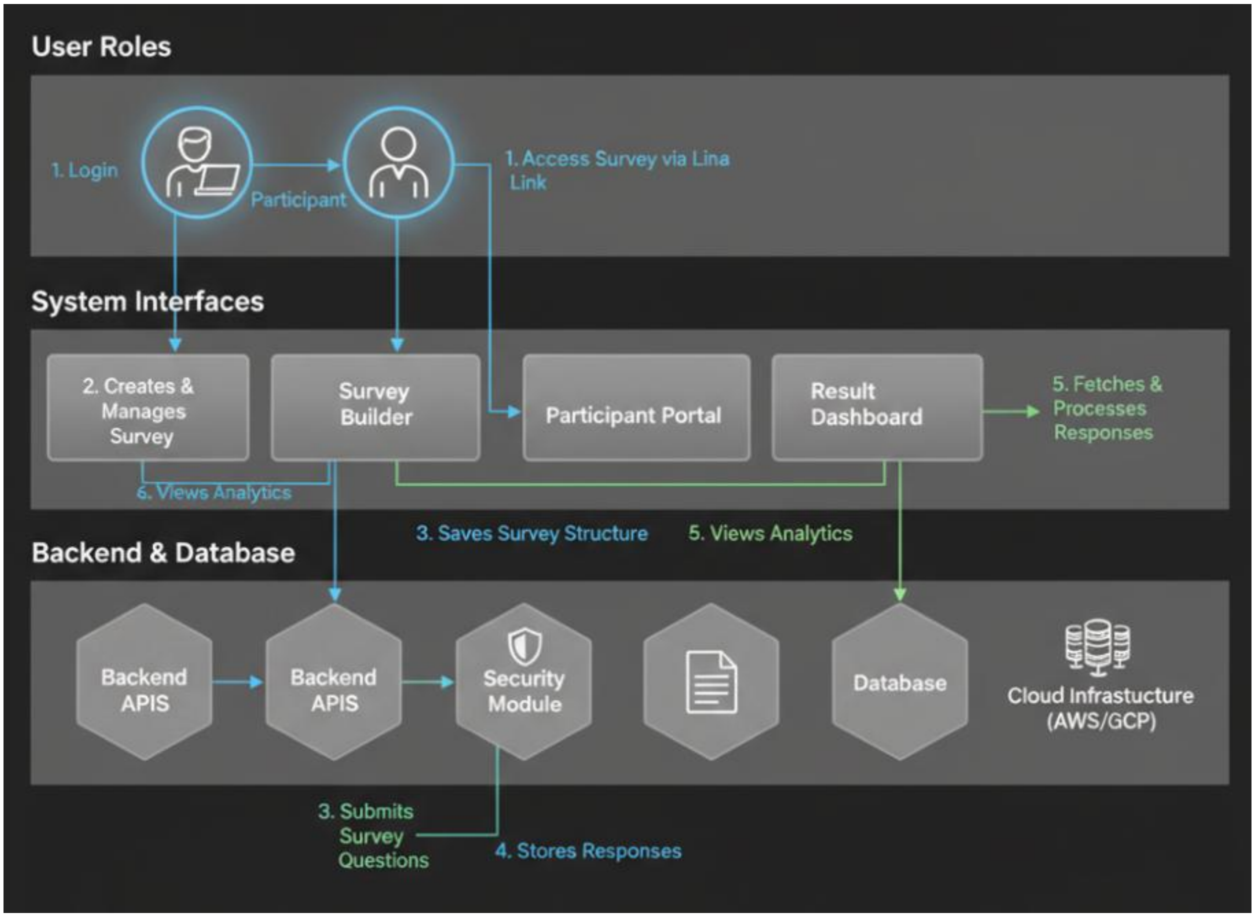
## 4. Security & Authentication Module

- Implement JWT-based authentication (login sessions).
- Role-based access control (Admin, Creator, Participant).
- Secure API endpoints with Spring Security.

- Input sanitization and validation to prevent attacks (e.g., SQL/NoSQL injection, XSS).

## 5. Integration & Deployment Module

- Connect frontend (React) with backend (Spring Boot APIs).
- Integrate backend with MongoDB Atlas (cloud-hosted DB).
- Add environment configs for development/production.
- Deploy application on platforms like Render / Vercel / Netlify + Heroku/Spring Boot server.



# Report on Module 1: Frontend – Survey Builder & Participant Portal (React.js)

## Introduction

Module 1 of the *Online Survey & Feedback System* is dedicated to the frontend development of two major components: the *Survey Builder UI* for survey creators and the *Participant Portal UI* for users submitting responses. This module is crucial for enabling smooth interactions between creators and participants, providing an intuitive, visually appealing, and responsive user interface. By utilizing *React.js*, the module benefits from a component-based architecture that ensures modularity, scalability, and maintainability. This approach allows developers to easily update or expand the system in the future while maintaining consistency across the interface.

The module also emphasizes accessibility, performance optimization, and responsive design to cater to a wide range of devices and screen sizes. It sets the stage for seamless integration with backend APIs and data storage modules, ensuring that user actions on the frontend translate efficiently into backend operations.

## Objectives

- To provide survey creators with a comprehensive, easy-to-use interface for designing, editing, and managing surveys.
- To allow participants to access, attempt, and submit surveys through a mobile-friendly and responsive portal.
- To implement dynamic rendering of surveys with multiple question types, including advanced options like conditional questions and branching logic.
- To establish a robust framework for connecting frontend interactions with backend APIs securely.
- To ensure a consistent and visually appealing UI/UX that promotes engagement and ease of use.
- To lay the groundwork for analytics and reporting in subsequent modules.

## Workflow

### Survey Builder UI (for creators)

1. **Survey Creation**:
   - Survey creators can build complex surveys using a variety of question types, including Multiple Choice Questions (MCQs), open-text fields, ratings, scales, and optional attachments.
   - Drag-and-drop functionality allows for flexible arrangement of questions, enhancing customization and workflow efficiency.
   - Real-time preview ensures creators can see the survey as participants would, allowing for immediate adjustments.
2. **Question Management**:
   - Creators can easily add, edit, reorder, or delete questions as needed.
   - Advanced options include mandatory field settings, character limits, and conditional branching.
   - Each modification can be previewed immediately to ensure accuracy before publishing.
3. **Publishing**:
   - Once finalized, surveys can be published, making them instantly available to participants in the portal.
   - The module also supports scheduled publishing and version control for ongoing surveys.

### Participant Portal UI (for participants)

1. **Survey Access**:
   - Participants can view a curated list of all available surveys, with search and filter capabilities by category, creator, or date.
   - The interface ensures quick navigation and easy selection of surveys of interest.
2. **Survey Response**:
   - Surveys are rendered in a responsive layout suitable for desktop, tablet, and mobile devices.

  - Validation checks prevent submission of incomplete responses and provide real-time feedback for correction.
  - Data is transmitted securely to the backend via APIs, ensuring privacy and integrity of participant responses.

3. **User Experience**:
    - A modern and clean design enhances usability and reduces participant friction.
    - Accessibility features such as keyboard navigation, screen reader support, and color contrast adjustments ensure inclusive participation.
    - Feedback mechanisms, such as confirmation messages and submission acknowledgments, enhance user trust and satisfaction.

## Technical Details

- **Frontend Framework**: React.js for dynamic and component-based UI development.
- **UI Libraries**: Material-UI and Tailwind CSS for responsive, customizable, and aesthetic designs; Formik for streamlined form handling.
- **State Management**: React Context API or Redux for efficient state handling across components.
- **API Communication**: Axios or Fetch for connecting frontend components with Spring Boot backend services.
- **Routing**: React Router facilitates seamless navigation between Survey Builder, Participant Portal, Preview, and other pages.
- **Form Validation**: Integrated client-side validation ensures data quality and improves user experience.

### Features Implemented

- Dynamic form rendering using reusable React components.
- Flexible survey builder supporting multiple question formats and conditional logic.
- Preview functionality to visualize surveys before publishing.
- Mobile-responsive design with adaptive layouts for all screen sizes.
- Robust form validation, error handling, and user feedback mechanisms.

- Secure API integration to ensure seamless data transmission between frontend and backend.
- Smooth user interactions with transitions and notifications to improve engagement.

## UI Flow Diagram

1. **Survey Builder Flow**:
   - Creator Dashboard → Create New Survey → Add/Edit Questions → Configure Settings → Preview → Publish → Monitor Responses.
2. **Participant Portal Flow**:
   - Participant Login → View Available Surveys → Select Survey → Complete Survey → Review → Submit → Confirmation.

## Expected Outcomes

- Streamlined and efficient survey creation process for administrators and creators.
- Increased participant engagement due to responsive and accessible user interfaces.
- Higher user satisfaction through modern UI/UX, intuitive navigation, and immediate feedback.
- Foundation for real-time analytics, reporting, and data visualization in later modules.
- Improved maintainability and scalability of the frontend system to support future enhancements.
- Enhanced data integrity and security by integrating validation and secure API communication.

## Conclusion

Module 1 establishes the essential frontend capabilities for both survey creators and participants. Leveraging React.js and modern UI frameworks, it delivers a dynamic, responsive, and highly usable interface. This module forms the foundation for the entire Online Survey & Feedback System, ensuring seamless interactions, secure data handling, and a scalable platform that can evolve with user needs and technological advancements.