SecureSense – Leveraging Human Behavior for Security and Building a Secure Digital Culture **(Mobile Application Platform)**

Project Proposal

Supervisor

Muhammad Salman

Submitted by

Umair Younus Khan

(2093-2020)

Abu Uzair

(1989-2021)

Department of Computer Science, Hamdard University,

Karachi.

01/07/2024

1. Introduction

In today's digital-first world, collecting and analyzing feedback is crucial for individuals, organizations, and businesses seeking to improve services, understand user preferences, or conduct research. Traditional methods of conducting surveys, such as paper-based forms or in-person interviews, are not only time-consuming but also lack scalability and efficiency. To overcome these limitations, modern digital survey platforms offer dynamic, fast, and accessible ways to gather information.

This project introduces a **Mobile-Based Survey Application**, designed to streamline the process of creating, managing, distributing, and attempting surveys. Built using **React Native** for cross-platform compatibility and **Node.js with MongoDB** for backend services, the app enables both **registered users** and **guest users** to interact with surveys based on their roles and permissions.

Registered users have full access to core features such as:

- Survey creation with customizable questions and options
- Survey editing, viewing, and deletion
- Viewing attempt results and scores
- Sending surveys via email
- Managing and reusing surveys

Guest users, on the other hand, can **only attempt a specific survey** using a unique survey ID and email. This limited-access approach allows organizations to target specific users while maintaining control over broader app functionality.

The app supports multiple types of questions such as multiple choice, and allows score calculation upon survey completion. All data is stored securely using **MongoDB**, and authentication is handled via secure tokens. To enhance performance and reliability, the frontend uses **async storage**, while backend logic follows a **REST API architecture**.

This mobile app simplifies survey management and offers a responsive, user-friendly interface with minimal learning curve. It serves as a practical tool for educational institutions, businesses, research organizations, and anyone needing quick and reliable survey feedback mechanisms.

By digitizing and automating survey workflows, this project not only increases data accuracy and collection speed, but also **empowers users** with insights through structured feedback, laying the groundwork for data-driven decision-making.

2. Objective

- **Design & Develop a Cross-Platform Survey Application**

Build a mobile app using React Native that works seamlessly both Android platforms, allowing users to interact with surveys intuitively.

- **Enable Role-Based Access**

Implement differentiated functionality for guest users and registered users. Guests can only attempt specific surveys, while registered users can create, edit, manage, and share surveys.

- **Support Survey Creation and Management**

Allow registered users to design surveys with multiple questions and options, edit existing surveys, delete them, or reuse them as templates.

- **Implement Survey Attempt and Result Calculation**

Enable users to attempt surveys and automatically calculate scores based on selected answers. Store and display results efficiently.

- **Facilitate Survey Sharing via Email**

Provide registered users the option to share survey links with others (including guests) via email using integrated SMTP functionality.

- **Ensure Data Storage and Secure Access**

Use MongoDB for backend data management and secure authentication for user access and session handling. Store basic user info locally using async storage.

- **Deliver Real-Time User Interaction and Feedback**

Offer responsive feedback during survey attempts and form submissions, improving user experience and minimizing errors.

- **Track Survey Attempts**

Maintain detailed logs of who attempted which survey and when, including for guest users via survey ID and email.

- **Promote Data-Driven Insights**

Provide a structured way for users (especially survey creators) to collect, analyze, and understand feedback data to make informed decisions.

- **Enhance Usability with a Simple UI**

Prioritize user experience with a clean, intuitive interface that allows users of all types to navigate, create, and attempt surveys with ease

3. Problem Description

In today’s data-driven world, collecting user feedback is essential for decision-making across industries—from education and research to business and governance. However, traditional methods of survey distribution and data collection—such as paper-based forms or basic web forms—are inefficient, error-prone, and lack scalability. These conventional approaches often lead to disorganized data, low response rates, and an inability to provide real-time insights.

One of the key challenges in survey management is **accessibility**. Many survey tools are web-based and may not offer smooth mobile access or offline functionality, limiting participation from users who prefer or rely solely on mobile devices. Furthermore, many platforms do not differentiate between **guest users** and **registered users**, creating challenges in controlling permissions and functionality based on user roles.

For organizations and individuals aiming to conduct quick and targeted surveys, another issue is the **lack of customization** and **reusability** of survey templates. In most tools, creating a survey from scratch each time consumes valuable time and limits productivity. Similarly, tracking the **results of individual attempts**, especially for surveys shared with guests, becomes difficult without a robust backend infrastructure.

Data privacy and secure submission of responses are additional concerns. Inadequate security measures can lead to unauthorized access, data leaks, or manipulation of survey results, which ultimately affects the integrity of the research or decision-making process.

Additionally, many available solutions are either overly complex or lack essential features such as:

- Score calculation based on selected answers
- Email-based sharing of surveys
- Real-time attempt logging
- Role-based access controls
- Offline usage or asynchronous data handling on mobile

These limitations highlight the need for a dedicated mobile solution that provides **end-to-end survey management**, supports both **registered** and **guest** workflows, and delivers **reliable, real-time survey data** collection and scoring.

This project aims to address these challenges by developing a **Survey Mobile App** using React Native for cross-platform accessibility and a Node.js + MongoDB backend to ensure scalability, real-time performance, and secure data management. By offering a user-friendly interface and clear separation of user roles, the app provides a modern and efficient way to create, manage, share, and attempt surveys anytime, anywhere.

4. Methodology

1. Project Planning

The project begins with detailed planning, which includes identifying the core functionalities of the survey app such as user registration, survey creation, attempt recording, and result calculation. The team set milestones for each development phase (e.g., design, development, testing), mapped tasks to Gantt charts, and held regular meetings with the supervisor to ensure smooth progress. Agile methodology was adopted for iterative development, allowing flexibility and constant feedback from test users.

2. Literature Review / Competitive Analysis

Before initiating development, an in-depth analysis of existing survey platforms (e.g., Google Forms, SurveyMonkey, Zoho Survey) was conducted to identify their strengths and limitations. This analysis guided the selection of unique features for our app — such as real-time scoring, guest access via email link, and mobile-first survey management. Literature review also included UI/UX best practices and backend architecture models suitable for scalable mobile apps.

---

3. User Requirement Analysis

Stakeholders (e.g., students, educators, business professionals) were surveyed to gather requirements. Based on their responses, key features were shortlisted:

- User authentication
- Guest login via survey ID
- Editable question bank
- Response tracking
- Secure storage of results
- Mobile responsiveness and easy navigation
  These insights drove the feature prioritization in early sprints.

---

4. Mobile App Design and Development

Using **React Native**, a fully functional mobile application was designed that works on both Android and iOS. Figma was used for wireframing and creating screen designs. Key screens included:

- Registration/Login
- Dashboard (Survey List)
- Survey Creator
- Guest Access Screen
- Attempt Screen
- Result Summary
  Each feature was broken down into modular components and reusable UI elements, making future upgrades easier.

---

5. Backend API and Database Design

The backend was developed using **Node.js with Express**, and **MongoDB** was used to store surveys, user profiles, and attempt logs. RESTful APIs were created to allow:

- Survey CRUD operations

- Guest-based survey access via ID
- Attempt submission and scoring logic
- Survey result calculation
  Secure routes and middleware ensured authentication and data validation.

---

6. Survey Logic and Scoring System

To handle objective-type questions, the app includes scoring logic where each option carries weight. On submission, the selected options are matched against correct ones, and a score is computed in the backend. The logic was implemented server-side to prevent manipulation and ensure integrity.

---

7. Testing and Feedback Cycle

The app was tested extensively through real-user scenarios:

- Registered user creates and shares a survey.
- Guest enters survey ID to attempt the survey.
- Submission is recorded, and score is shown.
  Manual test cases and automated Postman tests were used to validate APIs. A pilot group of users gave usability feedback, which was used to fix bugs and improve navigation.

---

8. Email Sharing and Guest Flow

An integrated email module (Nodemailer) was added for registered users to send survey links directly to guests. This module sends survey ID along with a friendly message and instruction set. Guests can access the survey without creating an account.

---

9. Documentation

Complete documentation was prepared including:

- API Documentation (Postman)
- App Usage Guide (with screenshots)
- Testing Logs and Feedback Summary
- Technical Flowcharts (for backend logic)
  This documentation helps both developers and end-users understand and operate the system efficiently.

5. Project Scope

Included in Project

The mobile-based Survey App is designed to streamline survey creation, distribution, and intelligent analysis through advanced features. The included modules are:

- **User and Guest Survey Access**: Registered users and guests can access surveys. Guests can enter a survey ID to attempt surveys without full registration.
- **Survey Creation and Management**: Logged-in users can create, edit, duplicate, delete, and share surveys with ease using a dynamic user interface.
- **Dynamic Survey Logic (Conditional Branching / Skip Logic)**: Surveys support skip logic, meaning questions are conditionally displayed based on user responses, allowing for smarter survey flow.
- **Real-Time Score Calculation**: Survey responses are automatically evaluated based on predefined correct answers, giving immediate feedback or results.
- **AI-Powered Response Interaction**: After submitting a survey, users can **ask follow-up questions (prompts)** based on their own survey responses. The **AI module** interprets survey answers and generates context-aware, meaningful insights (e.g., "Why did I score low in section 2?", "Which question was most important?").
- **Survey Sharing via Email**: Surveys can be shared with users or guests via email using integrated SMTP functionality.
- **Result Tracking and Attempt History**: Survey creators can view past attempts, scores, and trends to analyze survey effectiveness.
- **Cross-platform Mobile App**: Developed using React Native, compatible with Android and easily extendable to iOS.
- **Secure Backend Architecture**: Node.js, Express.js, and MongoDB are used for backend development, with RESTful APIs and secure data handling.
- **Modular Design and Scalable Codebase**: Designed with reusability, modularity, and future extensibility in mind.

Excluded from Project

To maintain a focused and manageable scope, the following features and areas are explicitly excluded from this project phase:

- **Offline Mode Functionality**: The app currently requires an active internet connection. Offline survey creation or attempt is not included in this release.
- **Multi-language Support**: Surveys and app content will only be in English. Localization and multi-language interfaces are not planned for this version.
- **Third-party API Integration (e.g., Google Forms, Typeform)**: The app is standalone and does not integrate with external survey platforms or services.
- **Advanced Analytics Dashboard for Admins**: Though basic results and attempt history are included, detailed visual analytics (e.g., charts, filters, export features) are not part of this initial scope.
- **User Role Management System**: There is no role hierarchy like Admin > Moderator > Surveyor. Only standard users and guests are supported.
- **Manual Review of Responses**: All survey evaluations and AI insights are automated. No manual grading, feedback, or moderation system is implemented.

- **AI Model Training or Customization by End Users**: The AI system uses a fixed backend model for interpreting responses. Users cannot train or customize the AI model in-app.
- **Voice Input or Speech-to-Text**: All survey inputs and AI prompts are text-based only.
- **Push Notifications or Reminders**: Users will not receive automatic reminders or notification alerts about surveys or results.
- **Legal Compliance Tools (e.g., GDPR Enforcement)**: While basic data protection practices are followed, the app does not implement a full legal compliance framework for regulations like GDPR, HIPAA, etc.

6. Feasibility Study

With the defined scope of the **Survey App with AI-based Insights**, the project is **technically and operationally feasible** within the allotted timeline, assuming effective risk management and resource availability. Below are the key feasibility components:

6.1 Risks Involved

1. **Technical Risks**
   Integrating real-time AI-based response generation and score evaluation increases complexity. Handling survey logic, backend scalability, and mobile device compatibility could pose risks if not properly tested.
2. **AI Logic Accuracy**
   The AI module, which responds to prompts based on user answers, must be context-aware. Ensuring relevant and ethical AI responses without hallucination is a key challenge.
3. **Skip Logic Exclusion Limitation**
   The absence of advanced conditional branching (skip logic) limits dynamic survey flow. While this simplifies development, it may impact user experience in complex surveys.
4. **Data Privacy Compliance**
   The app collects user responses, potentially including sensitive data. Ensuring secure storage, proper encryption, and data retention policies is crucial to avoid privacy issues.
5. **User Adoption Risks**
   Resistance to using digital platforms or mobile apps for surveys, especially among non-tech-savvy users or guest participants, could limit initial reach.

6.2 Resource Requirements

6.2.1 Computing Resources

- Cloud-based Node.js + MongoDB server.
- AI model host (e.g., OpenAI/GPT backend or custom model server).
- React Native build systems (Android Studio, Xcode, Expo).

- Backend testing tools (Postman, Insomnia).

6.2.2 Human Resources

- React Native Developer (Frontend)
- Node.js + Express Developer (Backend)
- MongoDB Admin (Database)
- UI/UX Designer
- QA/Test Engineer (Manual + Automation)
- AI Engineer (Prompt Engineering & API integration)

6.2.3 Other Resources

- Access to OpenAI API or similar LLM API provider.
- Hosting service (e.g., Heroku, Vercel, AWS).
- Firebase or local storage for guest login data.
- Documentation & version control tools (GitHub, Notion).

---

7. Application Areas

The AI-enabled survey app can be widely used across multiple domains:

- **Education**
  For conducting course feedback, online quizzes, or student satisfaction surveys.
- **Corporate Organizations**
  For gathering employee feedback, satisfaction surveys, performance evaluations, and exit interviews.
- **Healthcare**
  For patient satisfaction surveys, telemedicine follow-ups, or wellness check-ins.
- **E-commerce/Retail**
  Customer experience surveys after purchase or delivery.
- **Research & Development**
  Collecting user input for academic research, behavioral studies, or usability testing.
- **AI Feedback Testing**
  Useful in testing how users respond to AI-generated insights post-survey for behavioral evaluation.

---

8. Tools & Technology

8.1 Hardware Requirements

Component	Specification	Purpose
Development Laptop	i5/i7 Processor, 8GB RAM, SSD	Frontend, backend, and AI module development

Android Device	8GB RAM, 128GB Storage	Mobile testing and real-device deployment
Internet	4 Mbps download/upload minimum	API access, GitHub, and real-time syncing
Printer/Scanner	Laser Color + Scanner	For report printing or scanned documentation (if needed)

8.2 Software Requirements

Tool/Stack	Purpose
React Native	For mobile app development across Android/iOS
Node.js + Express	For backend server and REST API creation
MongoDB	NoSQL database to store user data, surveys, attempts
OpenAI API (GPT-4)	AI backend for responding to prompts based on survey attempts
JWT & bcrypt	User authentication and password hashing
Expo CLI / Android Studio	App build, testing, and deployment
Postman	API endpoint testing and debugging
VS Code	Development IDE
Git + GitHub	Version control and collaboration
Miro / Figma	UI/UX design and prototyping
Firebase (optional)	For guest login session storage and auth if needed
Power BI / Chart.js	For potential analytics visualization (in future)

9. Responsibilities of the Team Members

Team Member	Role	Responsibilities
Umair Younus Khan	Project Lead & Developer	**Project Planning and Coordination:** Develop project plans, timelines, milestones, and resource allocation.

		Risk Management: Identify risks, develop mitigation strategies, and monitor progress. **Stakeholder Communication:** Serve as the primary contact for stakeholders and ensure clear communication. **Quality Assurance:** Ensure all deliverables meet required quality standards. **Technical Architecture:** Design the SSBC Mobile App architecture. **Frontend Development:** Lead development of the user interface using React Native. **Backend Development:** Develop server-side components using Node.js and Express. **Database Management:** Set up and maintain MongoDB. **Code Review and Testing:** Conduct code reviews and execute test plans.
Abu Uzair	Research & Development Asistant	**Literature Review:** Conduct reviews of existing research in cybersecurity and behavioral compliance. **Survey Design:** Develop psychometric scales and questionnaires. **Data Analysis:** Analyze survey and behavioral data using SPSS, R, or Python. **Psychometric Analysis:** Perform psychometric analysis to validate scales. **Reporting:** Prepare reports and documentation on findings.

RACI Chart

Task/Activity	Umair	Uzair	Supervisor
Project Planning	**R, A**	**R**	**C, I**
Literature Review	**R, A**	**R**	**C, I**
Identifying Factors	**C. I**	**R, A**	**C, I**
Design Platform	**R, A**	**R**	**C, I**
Design Questionnaire	**R, A**	**R**	**C, I**
Integrate CybPsy Principals	**C, I**	**R, A**	**C, I**
Design Assessment Report	**R**	**R, A**	**C, I**
Evaluate Performance	**R, A**	**I**	**C, I**
Documentation	**A, I**	**R**	**C, I**

10. Planning

Task/Activity	Duration
Project Planning	2 weeks
Literature Review/Competitive Analysis	4 weeks
Identifying Factors	2 weeks
Design Database	6 weeks
Design & Development Platform	12 weeks
Integrate CybPsy Principals	8 weeks
Design Assessment Report	6 weeks
Evaluate Performance	4 weeks
Documentation	8 weeks
Total Estimated Duration	**52 weeks**

11. References

[1]. Mary Ann Liebert, Inc.

[2]. https://home.liebertpub.com/publications/cyberpsychology-behavior-and-social-networking/10/overview

[3]. Phys.org

[4]. https://phys.org/journals/cyberpsychology-behavior-and-social-networking/

[5]. Lee, C., & Park, J. (2023).

[6]. Behavioral Factors in Cybersecurity: Insights and Applications. Journal of Computer Science and Technology, 8(1), 22-35.

[7]. Williams, B., & Brown, M. (2024).

[8]. Design and Development of Mobile Applications for Cybersecurity: A Comprehensive Guide. Journal of Mobile Technology, 5(4), 78-92.