INFO-537 Project Report (Group - 4)

I have done this assignment completely on my own. I have not copied it, nor have I given my solution to anyone else. I understand that if I am involved in plagiarism or cheating, I will have to sign an official form that I have cheated and that this form will be stored in my official university record. I also understand that I will receive a grade of 0 for the involved assignment and my grade will be reduced by one level (e.g., from A to A- or from B+ to B) for my first offense, and that I will receive a grade of "F" for the course for any additional offense of any kind.

Name : -

Raj Barath Ezhilavanco, B01103748

Guru Pranav Narayanan, B01103149

Saroj Yadav, B01037179

Sadiq Shaik, B00984009

I. Git repository link for source code

https://github.com/Sarojv04/TDS-Project/tree/main

II. Used tools to create projects:

Visual Studio Code (VS Code):

- Used as the primary integrated development environment (IDE) for writing and debugging code. VS Code's extensions for Python and Django greatly streamline development, enabling features like syntax highlighting, IntelliSense, and debugging.

PostgreSQL:

- Chosen as the database to store survey data. PostgreSQL is robust and supports complex queries, making it ideal for managing survey details, user accounts, responses, and analytics.

Django Framework:

- Serves as the backend framework. Django is well-suited for developing web applications due to its robust ORM, security features, and built-in authentication system. Its modular structure allows the separation of concerns between models, views, and templates.

HTML and CSS:

- Used for designing the frontend interface. HTML structures the web pages, while CSS styles and enhances the user experience. Bootstrap (or similar CSS frameworks) may also be used to create a responsive design.

JavaScript:

- Employed for client-side interactivity, such as dynamic form validation, and enhancing the survey-taking experience. JavaScript frameworks or libraries like jQuery can simplify DOM manipulation.

Python (Object-Oriented Programming):

- Python's OOP principles are used to model real-world entities like surveys, users, and questions. It ensures maintainable and reusable code.

III. Design of the project:

- **User Authentication and Role Management**

- **Registration Page (Register.html):**
  This page allows new users to create accounts by entering a username, password, email, and selecting a role (Survey Creator or Survey Taker) via a dropdown menu. A "Register" button submits the form. Navigation to the login page is provided for existing users.

- **Login Page (Login.html):**
  This page supports user login with fields for username and password. Options like "Remember Me" enhance user convenience. A link to the registration page is included for new users.

- **Survey Creator:**

- **Dashboard (Creator_dashboard.html)**:
  After logging in, Survey Creators access their dashboard, which lists all surveys categorized as Drafts, Published, RePublished or Closed. Action buttons like "Edit", "Publish", and "Close" provide lifecycle management for surveys.

- **Survey Creation (Create_survey.html)**:
  Creators design surveys using a structured form with fields for survey name, description, and dynamically added questions. Each question includes a type dropdown (e.g., multiple-choice (Single Answer)) and option fields with "Add Another Option". Creators can save the survey as a draft or publish it directly.

- **Survey Management (Manage_surveys.html)**:
  A comprehensive view of all surveys with filter options by status. Creators can perform actions like editing or publishing surveys directly from this page.

- **Survey Editing (Edit_survey.html)**:
  Once the creator created a survey with all the questions and options in the survey create page. They can move that particular Survey to the Draft panel. And here they can edit the existing questions or options.

 Survey Analytics

- **Results Page (Survey_results.html)**:
  Here the Creator can view the survey results for all the panels. We have created a View results page for each panel like Published Panel, Republished Panel and Closed Panel.

  For example, Question 1 might show:
  - ✔ Option A: 20%
  - ✔ Option B: 40%
  - ✔ Option C: 30%

  This page provides a comprehensive view of survey performance and user responses.

 Survey Taker:

- **Survey List (Survey_list.html):**
  Survey Takers see a list of published surveys and also republished surveys if there are any, with titles, descriptions, and a "Take Survey" button as well as "Take Republished Survey" button. This ensures easy access to available surveys.

- **Survey Taking Page (Take_survey.html):**
  The survey form dynamically loads questions and options from the database. Users can select their answers and submit the survey with a "Submit" button.

- **Completion Message (Completion_message.html):**
  After submission, users see a confirmation message: "Thank You! Your survey has been submitted successfully." A navigation link redirects them to the survey list.

 Database Design (Table's Details):

  - **Users Table**: Stores user credentials and roles.

  - **Surveys Table**: Contains survey metadata, including name, description, and status.

  - **Questions Table**: Links to the survey, storing text and question type.

  - **Options Table**: Stores options for each question.

- **Responses Table**: Captures survey takers' responses.
- **Answer Table:** Capture survey answers by the users,for each question in the survey.

IV. Team Responsibilities:

1. Guru - Front-end Development:

Created and implemented user interfaces for all user and control pages, as shown in wireframes (Register.html, Login.html, Survey_list.html, Take_survey.html, Completion_message.html, etc…)

Ensure that displaying dynamic display searches based on status (published, republished, draft, or closed) works seamlessly with background logic.

Active survey forms, people can answer questions using radio buttons (i.e Multiple Choice with Single Answer), and enter data using Django and JavaScript templates.

Created an intuitive crowd intelligence framework, so that the survey takers can retake the survey from the republished panel. Where the Wisdom of crowd is implemented, i.e the takers can see the aggregated results for each question along with respective options.

Worked with Raj and Saroj to enable users to better integrate Crowd Data Intelligence into their user experience while maintaining clear insights.

2. Raj - Backend Logic and Survey Functionality (Backend Developer):

Developed basic logic for creating and managing surveys including managing different status (Draft, Published, RePublished, Closed) based on thread content and requirements.

Has built-in functions to collect survey responses and calculate the awareness of "Wisdom of Crowds" functionality.

Periodically handles study lifecycle events, including publishing studies to previously answered survey takers.

Queries are optimized for PostgreSQL to efficiently retrieve aggregated response data from the dashboard.

3. Saroj - User Authentication and Management Tasks (Project Administrator (Backend Developer)):

Implemented user authentication and registration using Django, allowing users to register as study creators or researchers.

Build an administrative dashboard in the admin panel for survey creators to enable them to create, manage, and publish surveys.

Used PostgreSQL database to store user data, survey information and responses to ensure safe and effective data storage.

The "Crowd Wisdom" function has been added to the backend to make it easier to publish questionnaires and collect preliminary responses.

4. Sadiq - Report:

Worked in hand with Saroj to generate the report for this project.Also worked with Raj while creating the wireframes for the web application pages.Which was then implemented through the front end development.

V. Project Challenges:

1) Dynamic Survey Question display: rendering the questions for the survey with the different types of options was a challenge in the beginning stages.

2) Republish Survey panel: After successfully implementing the survey panels and getting the data to reflect on the database, we needed to implement the republish functionality and it required a lot of changes in the front-end flow to achieve this result.

3) implementing the wisdom of the crowd was challenging because we were using a single table and trying to manipulate three different view results(publish result, republished result, view result) and it was a big task to get that done.

4) Admin Panel Design: Designing the admin panel where it will reflect the overall survey status and also handle the different stages of a survey(draft, publish, republish, closed) was challenging.

5) When the survey taker is taking a survey, we tried to view both published and republished surveys in a single page, which was very difficult as we needed to set the backend logic to achieve that specific goal.

6) One of the biggest challenges was to perfectly handle the surveys without removing them from the database and all stages of a survey had to be carefully planned before the implementation.

7) Survey Response count was a difficult task to implement(ie, to display the response count and the percentages of each option).

8) Properly handling errors when some user ‘s input was invalid was necessary and took some time figuring it out.

9) One of the greatest challenges we faced during the project was integrating the work of team members effectively. While one team member was focused on the front-end design, the other two were working on the back-end. And combining all the work to make the project run smoothly without any issue needed a lot of teamwork and collaboration.

VI. Conclusion:

Throughout the project, we worked as a team to solve key problems . Each member contributed equally:

Guru fixed the interface and provided powerful features, making the interface user-friendly and useful.

Saroj managed the database, user IDs, and feedback for analysis and project report creation.

Raj built powerful backend services for query management, change lifecycle, and feedback collection.

Sadiq worked with Saroj on the project report creation.

By collaborating effectively and leveraging everyone's strengths, we developed a collaborative research tool that achieved all of the project's goals, including the Wisdom of the Crowd concept.