|  |  |  |
| --- | --- | --- |
| **C:\Users\UIIT\Downloads\uaar logo.png** | **Pir Mehr Ali Shah**  **Arid Agriculture University Rawalpindi**  **University Institute of Information Technology** | **C:\Users\UIIT\Downloads\uiit logo.jpg** |

**FYP I – Comprehensive Survey Report**

Project Title: Constituency Connect

Supervisor: Ms. Farkhanda Qamar Project No:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Registration No.** | **Student Name** |
| 1 | 21-arid-737 | Muhammad Ahsan |
| 2 | 21-arid-719 | Huzaifa Bin Shahzad |
| 3 | 21-arid-715 | Haseeb Ur Rehman |

## Project Complexity:

### Multiple Roles:

The platform supports a complex multi-role system including.

* Constituents
* Representatives
* Assistants
* Department Complaint Handlers

Each role has distinct responsibilities, access levels, and functionalities, necessitating careful planning of user interactions and data management to ensure seamless and secure role-specific operations.

The interaction between these roles requires extensive user flow design and implementation, including complaint submission, tracking, and resolution processes, making the system significantly complex.

### Role-based Authorization:

The system uses a security method called JWT (JSON Web Tokens) to make sure users can only do what they're allowed to do.

Here's how it works:

1. **User Roles:**

* Each user is given a specific role
* Different roles have different permissions

1. **Authorization Process:**

* The system checks a user's role before letting them do something
* This helps keep the system secure

The complexity arises in defining and managing these roles while ensuring that the authorization mechanism is secure and efficient.

Implementing fine-grained access control for multiple roles requires robust backend logic and careful database schema design to enforce permissions and protect sensitive operations.

### Complex Data Handling:

The system manages multiple interconnected data entities, including:

* Users
* Complaints
* Surveys
* Events

Each entity has:

* Specific attributes
* Relationships with other entities
* A lifecycle with various events

### Authentication and Security:

Implementing industry-standard security measures is critical in protecting sensitive user data. This includes secure login mechanisms, password hashing, encryption of sensitive information, and protection against common threats like SQL injection, and XSS (Cross-Site Scripting).

The complexity also extends to ensuring that the security measures do not negatively impact the user experience, requiring a careful balance between security and usability.

### Real-Time Features:

The system requires real-time notifications and data synchronization across multiple devices, presenting two main challenges:

1. **Real-time Notification System:**

* Purpose: Instantly inform users about complaint status changes, event updates, and system alerts.
* Challenge: Integration with third-party services to create a robust, dependable, and scalable notification system.

1. **Multi-device Synchronization:**

* Purpose: Maintain consistent data across different user interfaces and devices in real time.
* Challenge: Preventing data inconsistencies while ensuring immediate updates on all platforms.

These features significantly increase the system's complexity, requiring careful design and implementation to ensure smooth operation and a seamless user experience.

### Performance Tracking and Data Visualization:

The system incorporates a comprehensive performance tracking mechanism for representatives, focusing on two key aspects:

1. **Performance Metrics:**

* Complaint resolution time
* Number of complaints managed
* Constituent engagement levels

1. **Data Processing and Visualization:**

* Complex data aggregation and analysis
* Generation of detailed performance reports
* Creation of visually appealing charts and graphs

This performance tracking system presents technical challenges in both back-end and front-end development:

* **Back-end:** Requires efficient algorithms for data processing and analysis to handle large volumes of performance data.
* **Front-end:** Demands advanced development skills to create interactive, visually compelling data visualizations that are both informative and user-friendly.

The goal is to provide meaningful insights into representative performance, enabling data-driven decision-making and continuous improvement of constituent services.

### User Experience and Interface Design:

The system needs a responsive interface that works well for all users. Making this interface easy to use for everyone requires careful planning and repeated testing. We need to make sure it's both useful and easy to access.

Another challenge in making the interface work smoothly on different devices is that this becomes tricky when dealing with:

* + Large amounts of data
  + Complex tasks like filling out forms
  + Real-time updates
  + Showing data in charts and graphs

### Customizable Reports:

The system offers customizable reports on various metrics, allowing users (e.g., representatives) to generate reports tailored to their specific needs. This involves complex data aggregation from multiple sources and the ability to filter, sort, and format data in various ways.

The backend must support dynamic query generation and efficient data retrieval to manage potentially large datasets, while the front end must provide a flexible and user-friendly interface for report customization.

### Multi-language Support:

To serve a diverse user base, the system offers support for multiple languages.

This includes:

* Translating the user interface
* Handling right-to-left (RTL) text for languages like Urdu and Arabic.
* Adapting date formats to match local customs
* Considering cultural nuances to ensure appropriate communication

This multi-language support adds complexity in several ways:

* Managing language files for each supported language
* Ensuring consistency in translations across different languages
* Providing a smooth user experience regardless of the chosen language

The end goal is to make the system accessible and comfortable to use for everyone, no matter their language preference.

## Technological Aspects:

### Frontend:

**HTML5:** The latest version of Hypertext Markup Language, used to structure the content on the web.

**CSS3:** The latest evolution of Cascading Style Sheets, used to style and layout web pages. CSS3 introduces new features like animations, transitions, and responsive design capabilities.

**Tailwind CSS:** A utility-first CSS framework that allows for rapid UI development with predefined classes, ensuring a consistent design and faster styling.

**Bootstrap:** A popular CSS framework that provides ready-to-use components and responsive grid systems, helping in quick and uniform web design.

**JavaScript:** A versatile programming language that powers the logic and interactive features on the web, enabling dynamic content and user interactions.

**React JS:** A JavaScript library for building user interfaces, particularly single-page applications (SPAs). It allows for efficient updates and rendering by using a Virtual DOM.

**Redux Toolkit:** A standardized way to manage and centralize application state in React apps, providing tools for easier state handling and debugging.

**Material UI:** A popular React component library that follows Google's Material Design guidelines, offering a modern and consistent UI/UX.

**React Hook Form:** A library that simplifies form handling in React, providing easy form validation and management with minimal code.

**React Router:** A standard library for routing in React, allowing navigation between different views and maintaining UI consistency.

**React Query:** A powerful tool for fetching, caching, and synchronizing server state in React applications, making API data management more efficient.

### Backend:

**Node JS:** A JavaScript runtime built on Chrome's V8 engine, enabling server-side scripting with JavaScript, providing a non-blocking, event-driven architecture ideal for scalable applications.

**Express JS:** A minimalist web framework for Node.js that simplifies server-side application development, providing robust features for web and mobile applications.

### Database:

**MongoDB:** A NoSQL database known for its flexibility in handling unstructured data, offering scalability and high performance for large-scale applications.

**Mongoose:** An Object Data Modeling (ODM) library for MongoDB and Node.js, providing schema-based data modeling and easy data validation.

### Authentication and Authorization:

**Clerk:** A modern user management solution that simplifies authentication and authorization, offering features like social login, authentication, and session management.

### Design and Prototyping:

**Visual Studio Code (VS Code):** A lightweight, versatile code editor with built-in support for debugging, syntax highlighting, version control, and a vast ecosystem of extensions. Ideal for efficient web development.

**Figma:** A cloud-based design tool used for creating user interfaces, wireframes, and prototypes collaboratively. Figma allows real-time collaboration and seamless handoff between designers and developers.

**Miro:** An online collaborative whiteboard platform used for brainstorming, mind mapping, and planning. Miro supports team collaboration in real time, enhancing project ideation and organization.

**Draw.io:** A diagramming tool used to create flowcharts, ER diagrams, and other visual representations. It helps in visualizing system architecture and processes, making it easier to communicate ideas.

**Adobe Illustrator:** A vector graphics editor used for creating detailed and scalable designs. Illustrator is ideal for creating logos, icons, and other graphical elements that require precision and flexibility.

### Why These Technologies?

* **React JS, Redux Toolkit, and related libraries:** These technologies are chosen for their efficiency in building dynamic and responsive single-page applications (SPAs), ensuring a smooth and interactive user experience.
* **Node.js and Express.js:** These tools provide a scalable and high-performance backend solution with asynchronous processing, ideal for managing multiple simultaneous user requests and ensuring efficient server-side operations.
* **MongoDB and Mongoose:** Selected for their flexibility in handling varied data structures and supporting scalable and fast data retrieval, which is crucial for the application's diverse data needs.
* **Clerk:** Chosen for its ease of use in user management, offering secure authentication and authorization with minimal setup, thereby streamlining the development process.
* **Figma, Miro, Draw.io, and Adobe Illustrator:** These design tools are used for thorough design, collaboration, and visualization, ensuring that the final product is well-rounded and polished.
* **Visual Studio Code (VS Code):** Used as the primary development environment for its robust features that support efficient code writing, debugging, and maintenance.

## Potential Impact on Society:

Constituency Connect aims to create a society where people are more connected to their representatives, government actions are more transparent, and the citizens are empowered to make their voices heard. This project has the potential to lead to better governance and a higher quality of life for everyone involved.

* **Improved Communication:** The Project will establish a direct communication route between constituents and their representatives. This will make it easier for people to express their issues and get them addressed. This improved communication will help foster trust and make people feel heard by those in power.
* **Increased Accountability:** Representatives will be more accountable to their constituents, as the system allows constituents to track the performance of their representatives.
* **Enhanced Civic Engagement:** The platform motivates citizens to take part in surveys, join virtual meetings, and give feedback on resolved complaints.
* **Efficient Problem Resolution:** The system’s ability to route complaints to the right department and prioritize urgent issues ensures quicker and more effective problem-solving.
* **Data-Driven Decision-Making:** Representatives can use the feedback and data collected through the system to make informed decisions that better serve their communities.
* **Empowered Citizens:** Constituents can stay informed about their representatives' activities and hold them accountable, leading to a more empowered and engaged citizenry.
* **Transparency in Governance:** By providing clear statistics and updates, the system fosters transparency in governance, helping to build trust between representatives and their constituents. Citizens can see how their representatives are performing and stay informed about the actions being taken in their community.

Benchmarking:

### **Problem:**

In many democratic societies, a significant communication barrier exists between elected officials and their constituents. Direct interaction with representatives is often limited to individuals with special connections or high social status, while the average citizen's interaction is typically limited to brief encounters during election seasons or at local events. These irregular interactions rarely address the ongoing needs and concerns of the community.

This lack of meaningful engagement has led to a sense of disconnection among citizens, who feel their voices are unheard in the democratic process. As a result, community issues frequently go unaddressed, and public trust in government institutions continues to diminish.

### **Existing Platforms:**

Although no such system exists in our local landscape that matches the scope of our project, some existing systems are similar to our initiative.

**FixMyStreet:**

FixMyStreet is a platform designed to help citizens report local issues, such as potholes, broken streetlights, and other community problems, directly to their local government. This user-friendly tool allows residents to track the progress of their reports, ensuring that public concerns are addressed efficiently and effectively.

**SeeClickFix:**

SeeClickFix is a community-driven platform that enables residents to report non-emergency issues in their neighborhood, such as damaged infrastructure, public safety concerns, and other local problems. This interactive approach encourages active community engagement and helps to build a stronger, more connected neighborhood.

**Neighbourland:**

Neighbourland is a civic engagement platform that empowers residents to collaborate with their local government and community organizations on neighborhood improvement projects. It focuses on gathering ideas, feedback, and support from the community to shape public spaces and policies. By facilitating open dialogue and collective action, it helps to create more vibrant, inclusive, and responsive communities.

### Comparison Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Features** | **Constituency Connect** | **FixMyStreet** | **SeeClickFix** | **Neighbourland** |
| **Origin** | Pakistan | United Kingdom | United States | United States |
| **Profile Management** | Yes, detailed profiles with editable information | Limited to basic profile management | Yes, basic profile management | Yes, basic profile management |
| **Complaint Reporting** | Yes, including anonymous reporting and detailed tracking | Yes, report local problems to the council | Yes, allows reporting issues to local authorities | No, focuses on community engagement |
| **Complaint Tracking** | Yes, with full history and status updates | Limited tracking, focus on reporting | Yes, with status updates | No, Does not focus on tracking |
| **Complaint Prioritization** | Yes | No | No | No |
| **Automated Complaint Routing** | Yes | Yes | Yes | No |
| **Virtual Meetups** | Yes | No | No | Yes |
| **Surveys and Polls** | Yes, Representatives can create and analyze survey | No | No | Yes |
| **Event Calendar** | Yes, Representatives can post upcoming events and public meetings | No | No | No |
| **Security** | Standard | Standard | Standard | Standard |
| **Feedback** | Yes, Feedback about problem resolution | Yes | Yes | Yes |
| **Performance Metrics** | Yes, tracks and displays the performance metrics of representatives | No | No | No |

## Project Features List:

Complaint Reporting, Complaint Forwarding to the respective department, Complaint Prioritization, Surveys and Polls, Real-time Analytics, Complaint Tracking Dashboard, Submission and Resolution, Statistics, Detailed Complaint View, Virtual Meetups, Video Conferencing, Anonymous Reporting, Confidential Submissions, Service Feedback, Event Calendar

## FYP Project Report Evaluation: (For Official use only)

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Good** | **Normal** | **Inferior** |
| Project Complexity |  |  |  |
| Technological Aspect |  |  |  |
| Potential Impact on Society |  |  |  |
| Benchmarking |  |  |  |
| Project Features |  |  |  |

**Suggestions/Remarks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Examiner |  | Date |  | Signature |

**Definition of Terms:**

* 1. **Project Complexity:** Project complexity is referred to as the degree of significant contribution that a group of students will make in the design and development of the project, spanning over two academic semesters. Secondly, determine if the domain of the project marks the standard of complexity required from a bachelor’s student degree, as this project will determine the skills they learned throughout the degree.
  2. **Technological Aspects:** Technological aspects of the project mean tools/technologies and language(s) used to develop it.
  3. **Potential Impact on Society:** Determine how much impact the product could have in its stated strategy for a society or community/focused group.
  4. **Benchmarking:** The proposed project should be compared with existing similar types of work. A ***comparison table*** is more helpful for a comparative view, listing features of existing works and proposed projects.
  5. **Project Features:** Verify that the features mentioned are complete and significant enough for an FYP project.