**Humber Admission System**

**Team Members:-**

**Rohan Balani: N01684022**

**Abhi Dhameliya: N01737293**

**Vrajeshbhai Dobariya: N01719087**

**Table of Contents**

Introduction …………………………………………………………………………………………………. 3

System Architecture ……………………………………………………………………………………… 4

Database ………………………………………………………………………………………………………. 4

Database scheme ………………………………………………………………………. 4

Entity relationship ……………………………………………………………………… 5

User manual …………………………………………………………………………………………………. 5

Views ……………………………………………………………………………………………………………. 7

Conclusion ……………………………………………………………………………………………………13

1. **Introduction**

* Managing the admission process in an educational institution can be rewarding, but without the right tools, it can quickly become a jumble of paperwork, scattered emails, and last-minute confusion. To address this, we have developed a user-friendly Student Admission System that ensures smooth and stress-free coordination for applicants, registrars, and administrators.
* Our application is built using Java with JavaFX for the interface and MySQL for the database. It is designed to support three main user roles:
  + - * Admins, who manage user accounts, oversee the entire system, and generate reports.
      * Registrars, who review applications, update applicant information, and make admission decisions.
      * Applicants, who can sign up, submit personal and academic details, and track their application status.
* With this system, admins can add, update, or remove users and maintain overall control, while registrars can view, evaluate, and manage applications in real time. Applicants can conveniently fill out their details, choose preferred programs, and submit their applications with just a few clicks.
* Every component of this application is designed with real-world needs in mind: secure authentication and role-based access for user control, an intuitive interface for ease of use, and robust error handling to maintain smooth and reliable operation.

1. **System Architecture**

* The system follows a three-layer architecture:

1. **Presentation Layer (UI)** – Built with JavaFX. Includes login, signup, applicant management, registrar dashboard, and admin dashboard screens.
2. **Business Logic Layer** – Contains controllers that handle application logic, validations, and role-based permissions.
3. **Data Access Layer (Persistence)** – Uses Hibernate ORM and JDBC for database operations.
4. **Database**

* **Database Type:** The system uses MySQL as the primary relational database.
* **Access Method:** Data access is handled through Hibernate ORM and JDBC, enabling seamless object-relational mapping between Java classes and database tables while reducing the need for extensive manual SQL coding.
* **Benefits:** This approach ensures strong referential integrity for maintaining consistent relationships between entities, allows highly optimized querying for efficient report generation, and provides a scalable structure capable of supporting future feature expansion and increasing data volumes without compromising performance.

**Database Integration (Implemented)**

**Database: MySQL  
Access: Hibernate ORM + JDBC  
Purpose: To store, secure, and manage admission process data for Applicants, Registrars, and Admins.**

**Table: stu\_applicants**

**Stores all applicant details and program preferences.  
Columns:**

* **applicant\_id *(PK, Auto Increment)* – Unique applicant ID.**
* **full\_name – Full name of applicant.**
* **username – Unique login name.**
* **password – Encrypted password.**
* **phone\_number – Contact number.**
* **email – Email address.**
* **status – Application status *(Pending, Under Process, Accepted, Rejected, Conditionally Accepted)*.**
* **program\_1, program\_2, program\_3 – Chosen programs (up to 3).**
* **is\_submitted – Boolean (0 = Draft, 1 = Submitted).**
* **created\_at, updated\_at – Record timestamps.**

**Table: sys\_admins**

**Stores system administrators with full system privileges.  
Columns:**

* **admin\_id *(PK, Auto Increment)* – Unique admin ID.**
* **full\_name – Admin’s full name.**
* **username – Unique login username.**
* **password – Encrypted password.**
* **phone\_number – Contact number.**
* **email – Email address.**
* **created\_at, updated\_at – Record timestamps.**

**Table: sys\_registrars**

**Stores registrar details who manage applicant records.  
Columns:**

* **registrar\_id *(PK, Auto Increment)* – Unique registrar ID.**
* **full\_name – Registrar’s full name.**
* **username – Unique login username.**
* **password – Encrypted password.**
* **phone\_number – Contact number.**
* **email – Email address.**
* **created\_at, updated\_at – Record timestamps.**

**Integration Highlights**

* **Hibernate maps these tables to Java entity classes: Applicant, Admin, Registrar.**
* **JDBC used for custom SQL queries and CRUD operations.**
* **Role-based access enforced by linking login credentials to the correct table.**
* **Data separation between roles ensures higher security and easier maintenance.**

1. **Entity Relationship**

In the Humber Admission System, all operations revolve around three main entities — **Admins**, **Registrars**, and **Applicants**.  
Each entity type has its own dedicated table (sys\_admins, sys\_registrars, stu\_applicants) and specific permissions within the system:

* **Administrators** can add, update, or remove registrars and applicants, oversee all application data, and generate reports.
* **Registrars** review and update applicant information, manage application statuses, and generate program-specific reports.
* **Applicants** can register themselves, submit personal and academic details, select up to three preferred programs, and track their application status.

Since all user roles are stored in **separate tables**, the role is determined by which table the user’s credentials are stored in rather than a shared “roles” table.

The **stu\_applicants** table holds all application data, including program preferences, submission status, and timestamps. Application status is updated directly in this table, eliminating the need for a separate status\_log table.

**Relationships in the System**

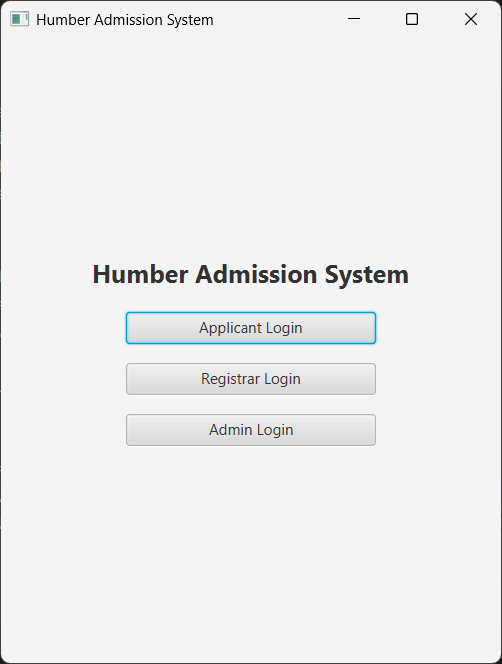
* **Admins → Registrars**: An admin can manage multiple registrars.
* **Registrars → Applicants**: A registrar can manage and process multiple applicants.
* **Applicants → Programs**: Each applicant can select up to three program fields (program\_1, program\_2, program\_3).
* **Applicants → Status**: Status changes (Pending, Under Process, Accepted, Rejected, Conditionally Accepted) are stored in the applicant record itself.

1. **User manual**

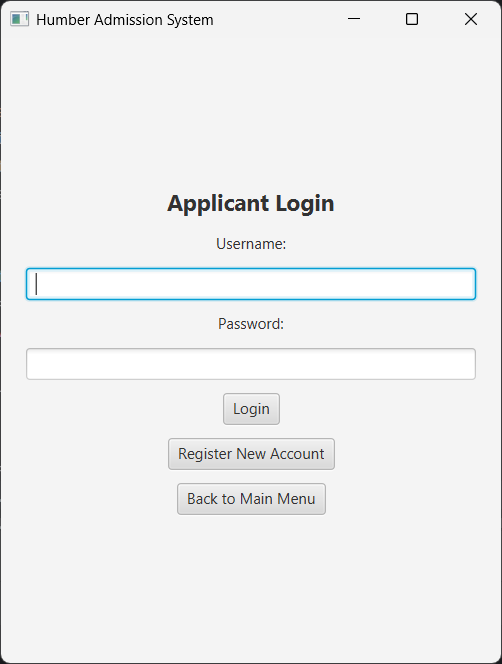
* **Logging In:**
* When the Student Admission System launches, users enter their username and password. After successful authentication, they are redirected to a dashboard tailored to their role.
* **Administrator Features**
* Add, update, or remove registrar accounts.
* View and manage all applicant records.
* Generate system-wide admission reports.
* **Registrar Features**
* Add, update, or remove applicant information.
* Review applications and update statuses (Pending, Under Process, Accepted, Rejected, Conditionally Accepted).
* Generate reports based on program, date range, or applicant criteria.
* **Applicant Features**
* Sign up and create a new application profile.
* Enter personal and academic details.
* Select up to three preferred programs.
* Track application status in real time.
* **Implementation Details**
* **Language Used:** Java (JavaFX)
* **Database:** MySQL (accessed via Hibernate ORM and JDBC)
* **Security:** Password-based authentication with role-based access control ensures that each user can only access features permitted by their role.
* **Error Handling:** The system provides clear and descriptive error messages to guide users when something goes wrong.

1. **Views**

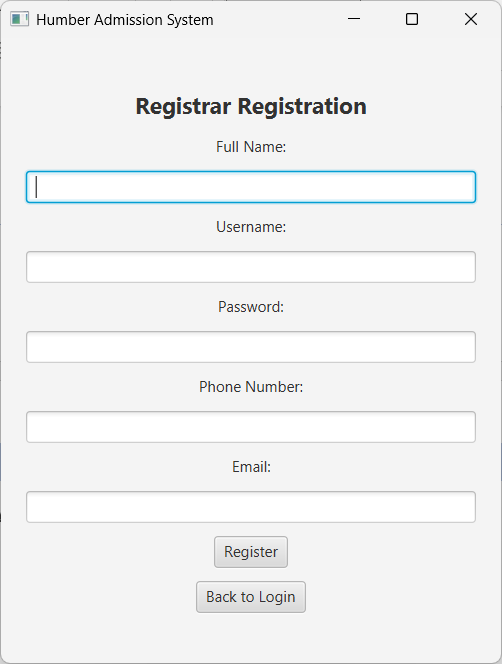
- **Welcome Page**



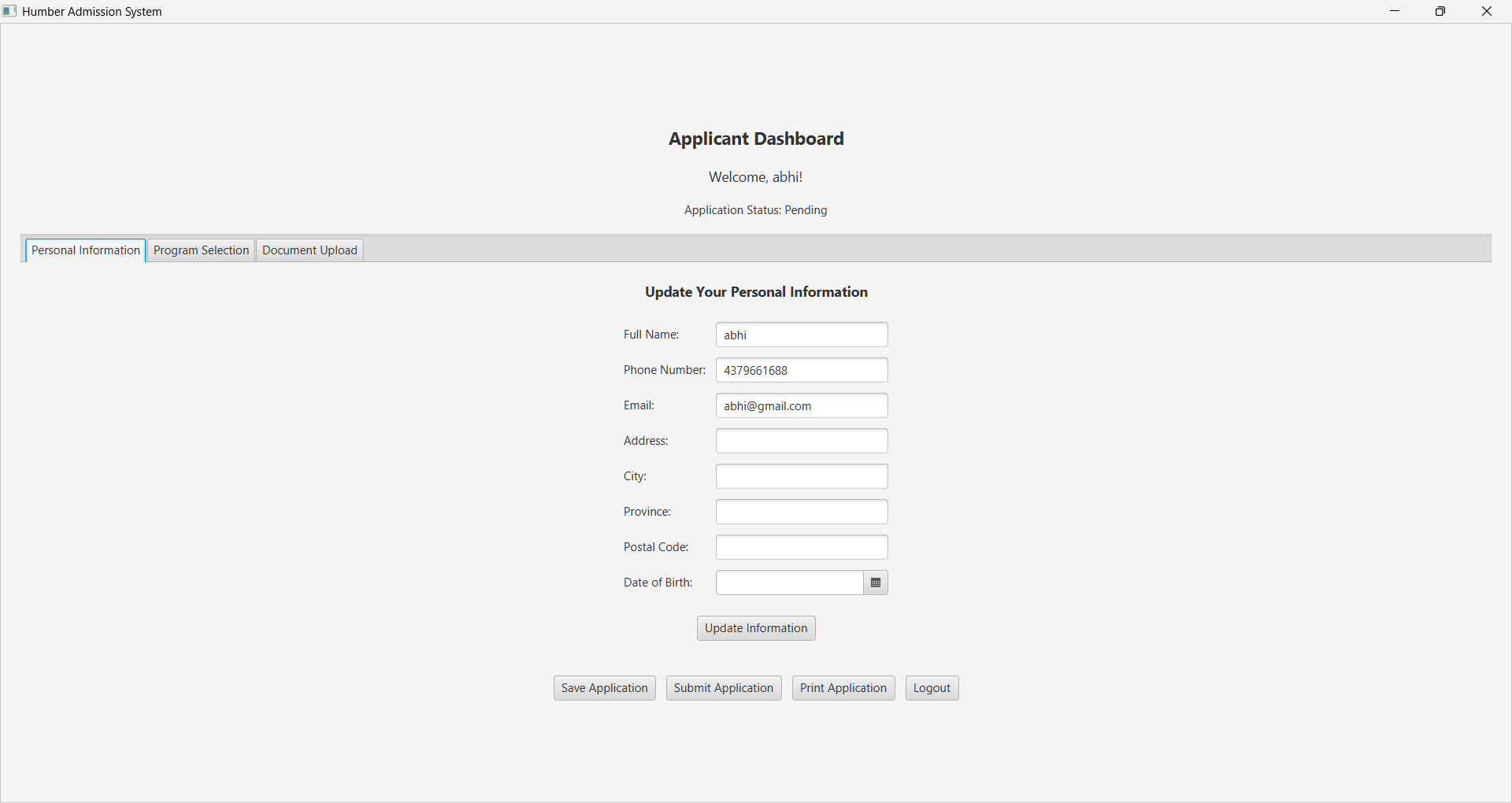
* **Login page (Applicant / registrar / admin)**

****

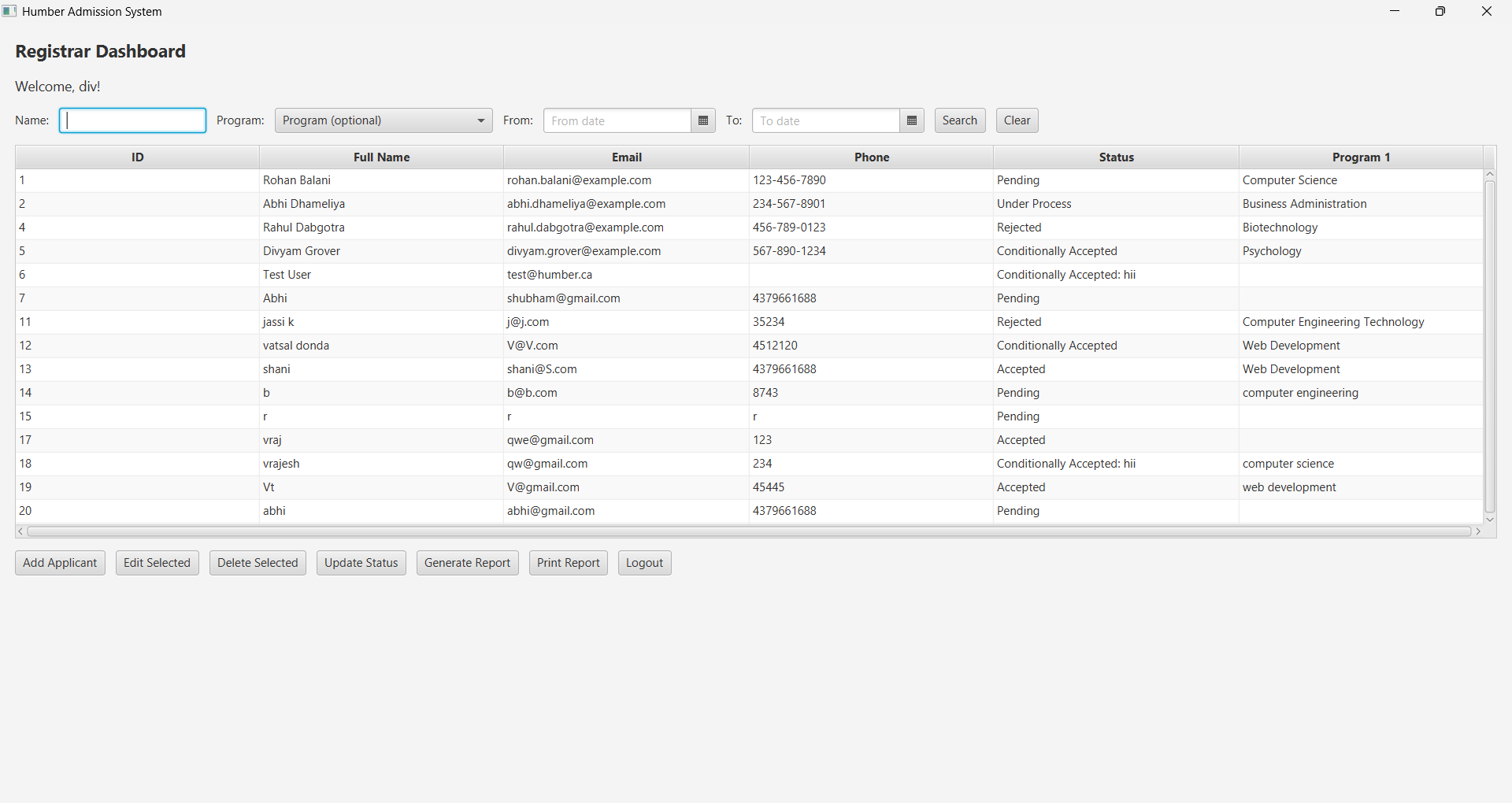
* **New Account Register page (applicant / registrar)**

****

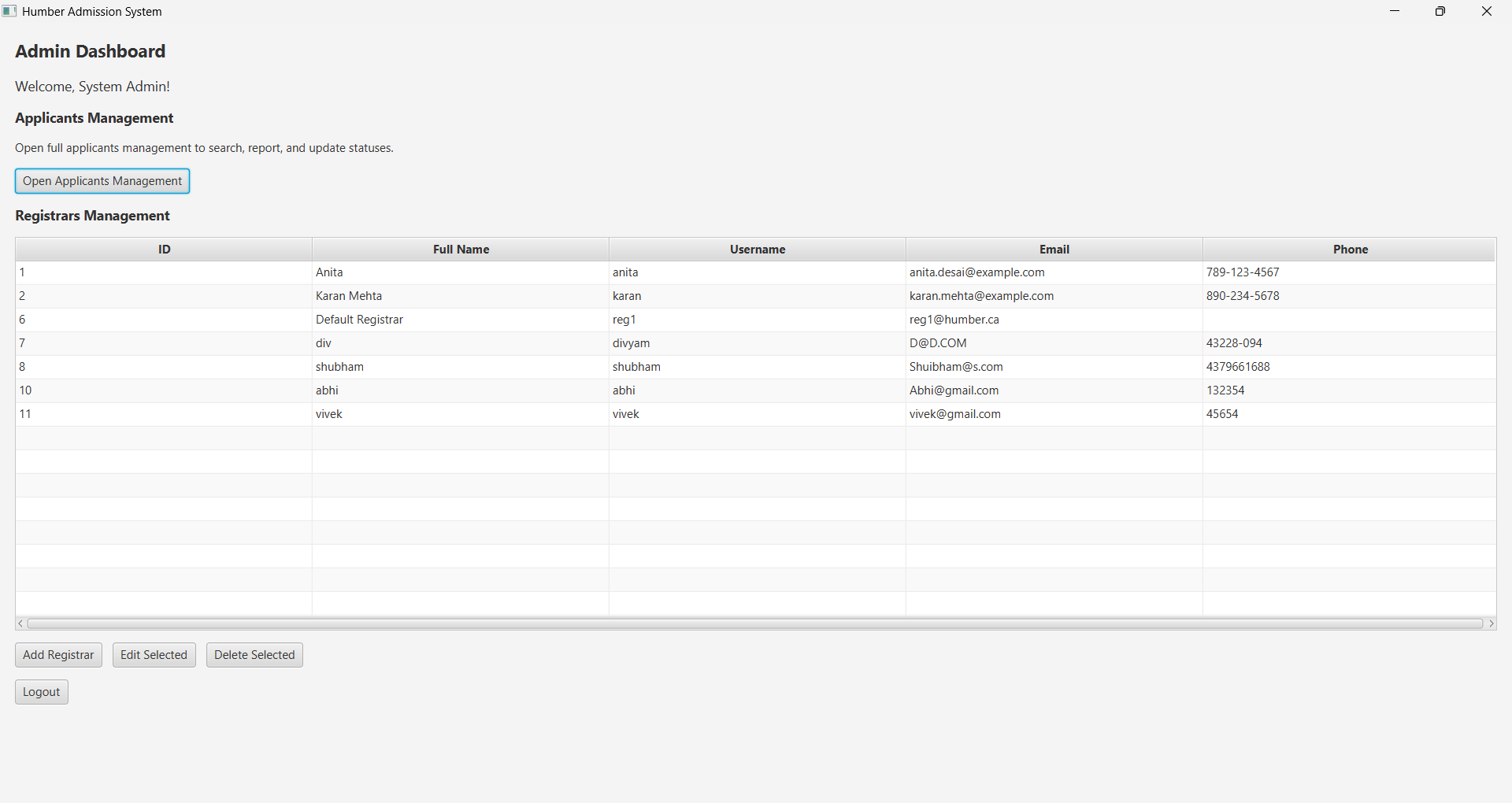
* **Applicant dashboard**

****

* **Registrar Dashboaard**

****

* **Admin Dashboard**

****

1. **Conclusion**

* The **Humber Admission System** integrates applicant, registrar, and administrator workflows into a single streamlined platform. It reduces manual paperwork, improves application tracking accuracy, and provides valuable reporting insights to support data-driven decisions. Its modular design makes the system easy to maintain and expand, allowing for future enhancements such as online document verification, integrated payment processing, and automated notification systems. This ensures the platform remains adaptable to evolving institutional requirements while supporting efficient and transparent admission processes for long-term success.