ID: 23445

NAMES: GIRAMATA ARLENE

STUDENT REGISTRATION SYSTEM

PROJECT REQUIREMENTS

THE PURPOSE OF THE PROJECT

Automate the processes related to student information, it will help to manage the most important processes for student management which include student admission management as well. The admission forms the backbone of a school and helps to strengthen the economy as well.

EXPECTED OUTCOME

- Manages school admission
- Manages student management

SPECIFIC CONSTRAINTS OR LIMITATIONS

- The system is not able to download the admission form of the student, after they are done filling it.

FUNCTIONAL REQUIREMENTS

- The user shall be able to sign up to the system
- The user shall be able to login to the system
- The user shall be able to view the registration form
- The user shall be able to upload the photo
- The user shall be able upload the curriculum vitae
- The user shall be able to update their information on the registration form
- The user shall be able to delete their registration form in case, in case they decide not to join the school
- The system shall be able to save the signup information in the database
- The system shall be able to retrieve data in the database to login
- The system shall be able to save the information from the registration form
- The system shall be able to send an email to the admin for the new user who registered himself/herself.

NON-FUNCTIONAL REQUIREMENTS

- The system should allow a user who signed up to login
- The user should fill the admission form only if they have logged in.
- The system shall be able to respond in 5 seconds
- The system shall be able user friendly
- The system code must be easy to read and understand, with clear naming conventions, comments, and documentation.
- The system must be developed following best practices for code maintainability, such as using design patterns, avoiding code duplication, and keeping code complexity to a minimum.
- The system must be able to perform its functions efficiently and without errors under different load conditions, such as high traffic or peak usage.
- The system shall able to run on window 10 or window 11 without change in its behavior.

PROJECT PLAN

PROJECT SCOPE

- Student Information Management:
  - Registration and enrollment details
  - Personal Information (First name, Last name, Email)
- Scalability and performance:
  - Design the architecture that can handle a growing number of students
  - Perform optimization to ensure efficient system response
- User roles and permission:
  - Role-based access control and permission management
  - Security measures to project student data and privacy
- Reporting and Analytics
  - Compliance reporting

PROJECT PLAN

- Project Initiation Phase (1 day)
  - Define project objectives,
  - Define scope
  - Define requirements.

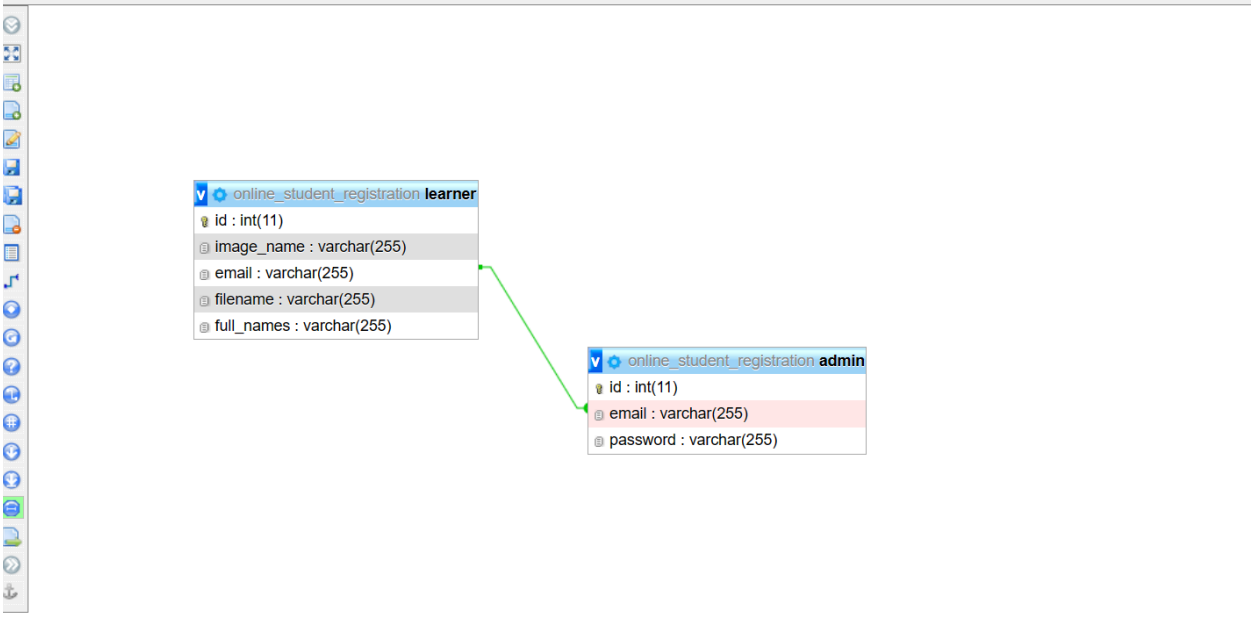
- Develop a high project level plan

- Requirements gathering and analysis phase (1 day)
  - Conduct detailed requirements gathering sessions
  - Analyze existing processes and systems to identify gaps and improvement opportunities.
  - Document functional and non-functional requirements
  - Prioritize requirements based on their importance and feasibility.
- System Design Phase (1 day)
  - Develop a detailed system architecture and design
  - Design Database Schema
  - Create wireframes and user interface mockups
- Development Phase (7 days)
  - Implement the student management system based on the approved design
  - Develop module for student registration
  - Conduct regular sprints and iterations
  - Perform unit testing to ensure functionality and quality of individual components
- Testing Phase (1 day)
  - Conduct system testing to verify end-to-end functionality of the system
  - Perform integration system to ensure smooth interactions between the module
  - Identify any identified issues or bugs
- Deployment (1 day)
  - Prepare the production environment for the deployment
  - Install and configure the student management system in the production environment

RESOURCES USED IN THE PROJECT

- Project Team
  - Project manager
- Software and Tools
  - Integrated Development Environment (IDE): Visual Studio Code
  - Database Management System (DBMS): MySQL
  - Version Control System: Git
- Training and Documentation:
  - Training Materials: guides and training videos
  - Documentation Tools: Google Docs for creating project documentation

DATABASE SCHEMA



USER DOCUMENTATION

The purpose of student management system is to help the student as well as the school fill the admission form digitally not manual on the paper but instead using a system where the student's data will be kept safely and for a long period of time.

When the user goes to the first page, he finds the home page that has the buttons for about, service, Sign Up, log In.

When you click on the About button, it brings the About page that contains the description of the site and what it is all about.

When you click on the Service Button, it brings The Service page contains the detailed description of the services that you can find on our site.

When you click on the Signup button it brings the Signup page which contain the Signup form where you can fill two credentials which are email and password.

When you click on the Login button, it brings the Login page, which contains the Login form where you can fill two credentials which are email and password, as the ones you used while signing up.

After logging in the system directs you to the admission page which has the admission form where the user should fill the credentials.

The credentials that should be filled on the admission form is full names, email, curriculum vitae, passport photo.

When the user is done and submit the form, the system will send an email to the admin, telling the admin that someone sent the admission.

TECHNICAL DOCUMENTATION

- Architecture
  - The student registration systems follow the Model-View-Controller (MVC) architectural pattern
  - The backend is implemented using Java programming language and Spring boot framework
  - The frontend is developed using HTML, CSS and JavaScript
- Development Environment
  - Operating system: Windows 10
  - IDE: Visual studio code
  - Database: MySQL
- Backend Implementation
  - The backend is implemented using Java and Spring boot
  - The project is organized into different packages such as controllers, services, models and repositories.
  - Data persistence is achieved using HTML, CSS and JavaScript
  - Input validation and error handling are implemented using Spring validation and exception handling mechanisms
- Frontend Implementation
  - The frontend is developed using HTML, CSS and JavaScript
  - Thyme leaf template engine is used for server-side rendering of dynamic webpages
  - Bootstrap is used for responsive
  - Client-side form validation is implemented using JavaScript
- Database Implementation
  - The database is implemented using MySQL
  - The database Schema is used to store student information and their login credentials
- Security
  - User authentication and authorization are implemented using spring security
  - Access control is enforced based on user role and permission
- Deployment
  - Configuration files such as application.properties is used to specify database connection settings and other system properties.
- Testing
  - Integration testing is conducted to test the interaction between different pages of the system.
- Logging and Error handling

- Custom error page or exceptions handlers are implemented to provide a user-friendly experience in case of errors.
- Errors and exceptions are logged with appropriate severity and error messages.

❖ Maintenance
- The system is maintained using version control systems such as Git to track changes.
- Regular updates and bug fixes are performed to ensure system stability and security.
- System documentation is updated to reflect any changes or enhancements made to the system.