ACADEMIC REQUEST MANAGEMENT SYSTEM CASE STUDY

INTRODUCTION

- The Academic Request Management System is a web-based application developed to simplify and automate the process of submitting and approving academic-related requests within an educational institution.

- Traditionally, students submit requests manually, which leads to delays, lack of transparency, and paperwork overhead. This system provides a centralized digital platform where students can raise requests, faculty members can review and take action, and administrators can monitor the overall workflow efficiently.

- The system is designed using Spring Boot for backend development, Thymeleaf for frontend rendering, and MySQL as the database. Role-based access control ensures that each user interacts only with the functionalities assigned to their

ABSTRACT

- The Academic Request Management System is a web-based application for managing student academic requests digitally.
- It allows students to submit requests and track their approval status.
- Faculty members can review and approve or reject requests.
- Administrators can monitor all requests within the system.
- The system uses Spring Boot, Thymeleaf, and MySQL.
- Role-based access ensures security and data integrity.

OBJECTIVES OF THE SYSTEM

- To digitize the academic request submission process
- To reduce paperwork and manual tracking
- To provide transparency in request status
- To enforce role-based access (Student, Faculty, Admin)
- To prevent modification of requests after approval or rejection

SCOPE OF THE PROJECT

- The scope of this project is limited to managing academic requests within an institution.
- The system supports three user roles: Student, Faculty, and Admin. Students can submit requests, faculty can approve or reject them, and administrators can view all requests for monitoring purposes.
- The project does not include external integrations such as email notifications or mobile access.

SYSTEM MODULE

Student Module

- Student login and authentication
- Submission of academic requests
- Viewing request status (Pending / Approved / Rejected)
- Viewing own profile details

Faculty Module

- Faculty login and authentication
- Viewing student requests along with student name
- Approving or rejecting requests
- Restriction on modifying a request after action

Admin Module

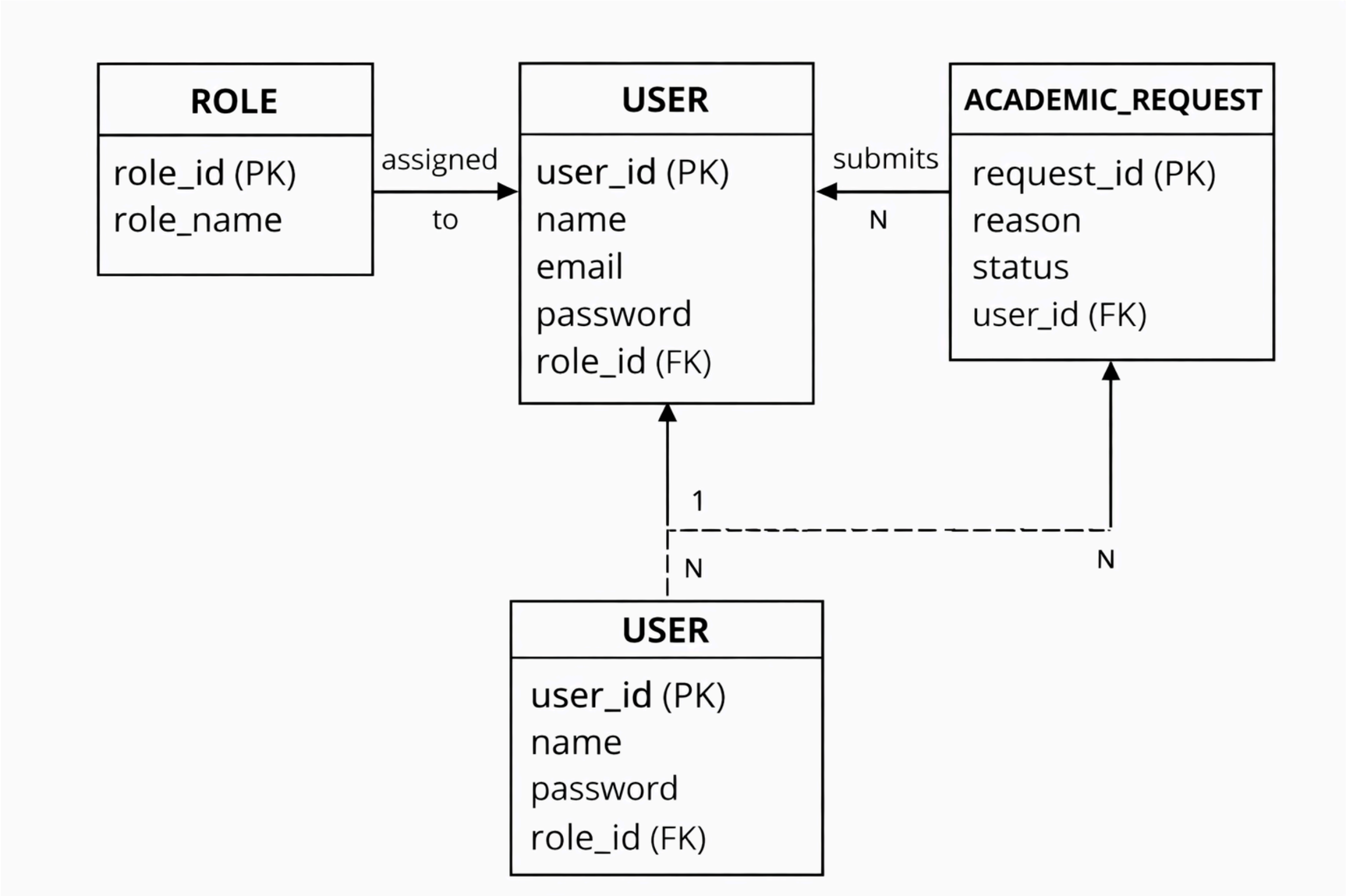
- Admin login and authentication
- Viewing all academic requests (read-only)
- Monitoring overall system activity

SYSTEM ARCHITECTURE

The system follows a layered architecture consisting of:

- Presentation Layer (Thymeleaf HTML pages)
- Controller Layer (Spring MVC Controllers)
- Service Layer (Business Logic)
- Repository Layer (JPA Repositories)
- Database Layer (MySQL)

Each layer is designed to ensure separation of concerns and better maintainability.


ROLE
role_id (PK)
role_name
assigned
to
USER
user_id (PK)
name
email
password
role_id (FK)
submits
N
ACADEMIC_REQUEST
request_id (PK)
reason
status
user_id (FK)
1
N
N
USER
user_id (PK)
name
password
role_id (FK)

ENTITY RELATIONSHIP DIAGRAM (ER DIAGRAM)

The Entity Relationship Diagram represents the logical structure of the database and the relationships between different entities in the system.

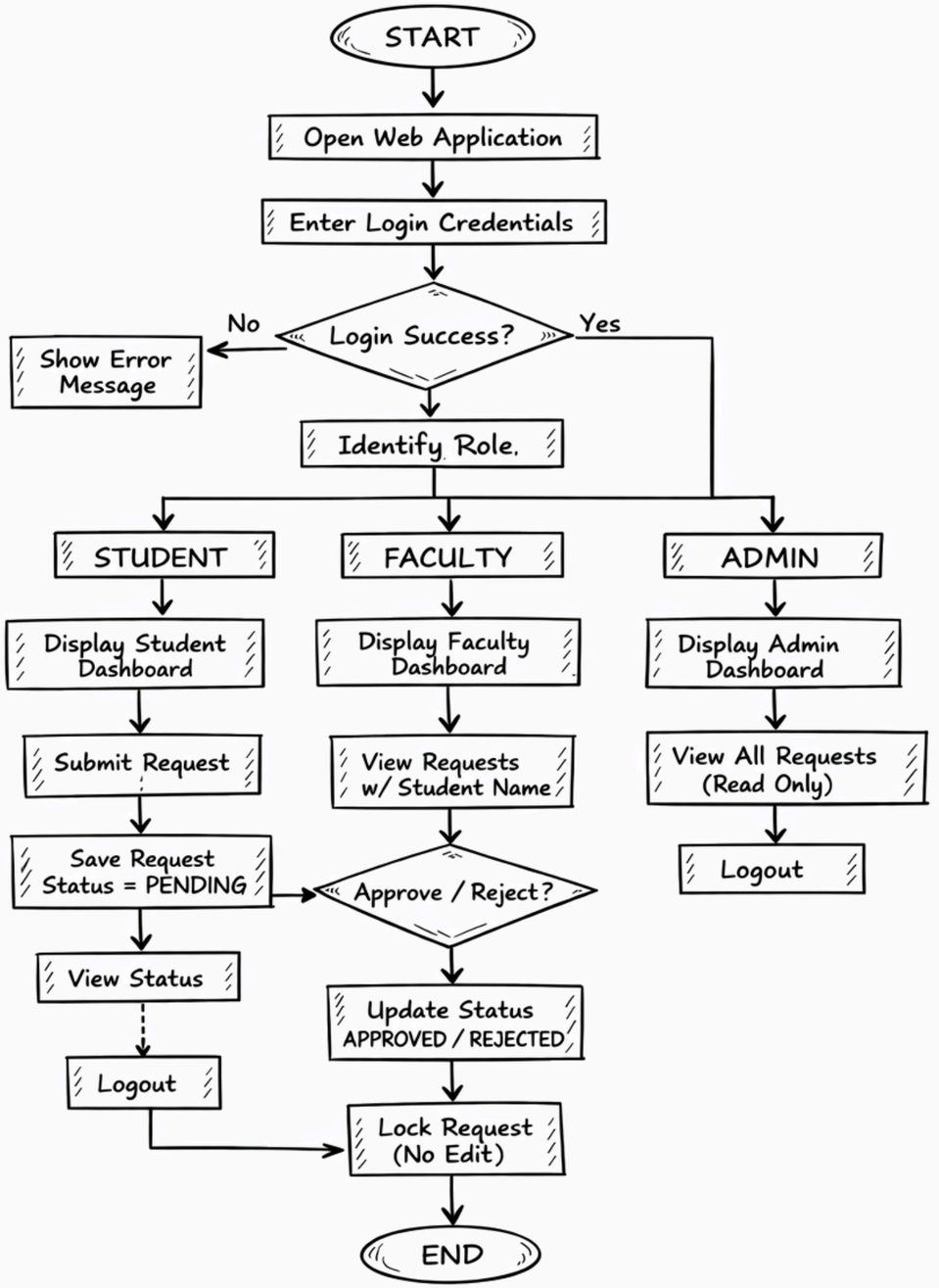
Entities:

- User
- Role
- AcademicRequest

Description:

- Each User is assigned one Role
- A Student (User) can submit multiple Academic Requests
- Each Academic Request is associated with exactly one Student
- Request status determines whether it is Pending, Approved, or Rejected

Academic Request Management System



Flow Description:

- User opens the application
- User enters login credentials
- System validates credentials
- If login fails, an error message is displayed
- If login succeeds, user role is identified
- Student can submit academic requests
- Faculty can approve or reject requests
- Admin can view all requests
- Once a request is processed, it is locked
- User logs out and the process ends

Technology	Description
Java	Backend programming language
Spring Boot	Backend framework
Spring Security	Authentication and authorization
Thymeleaf	Frontend template engine
MySQL	Database
Maven	Dependency management
GitHub	Version control

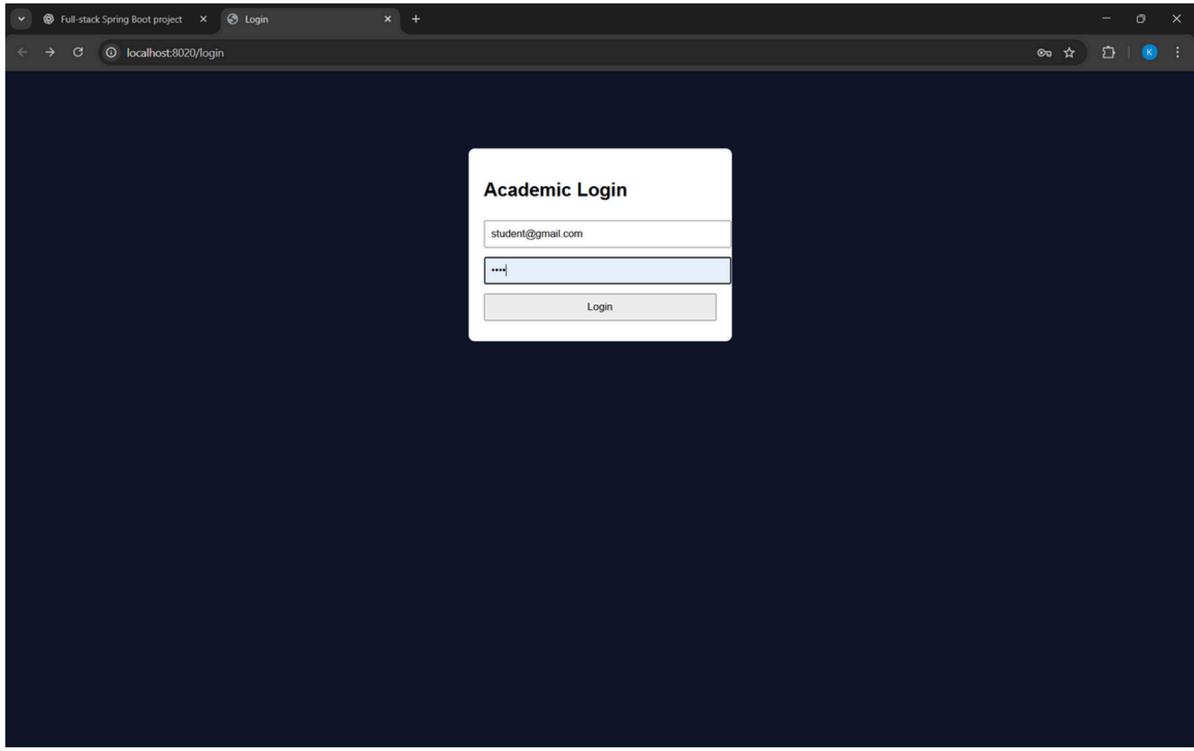
CONCLUSION

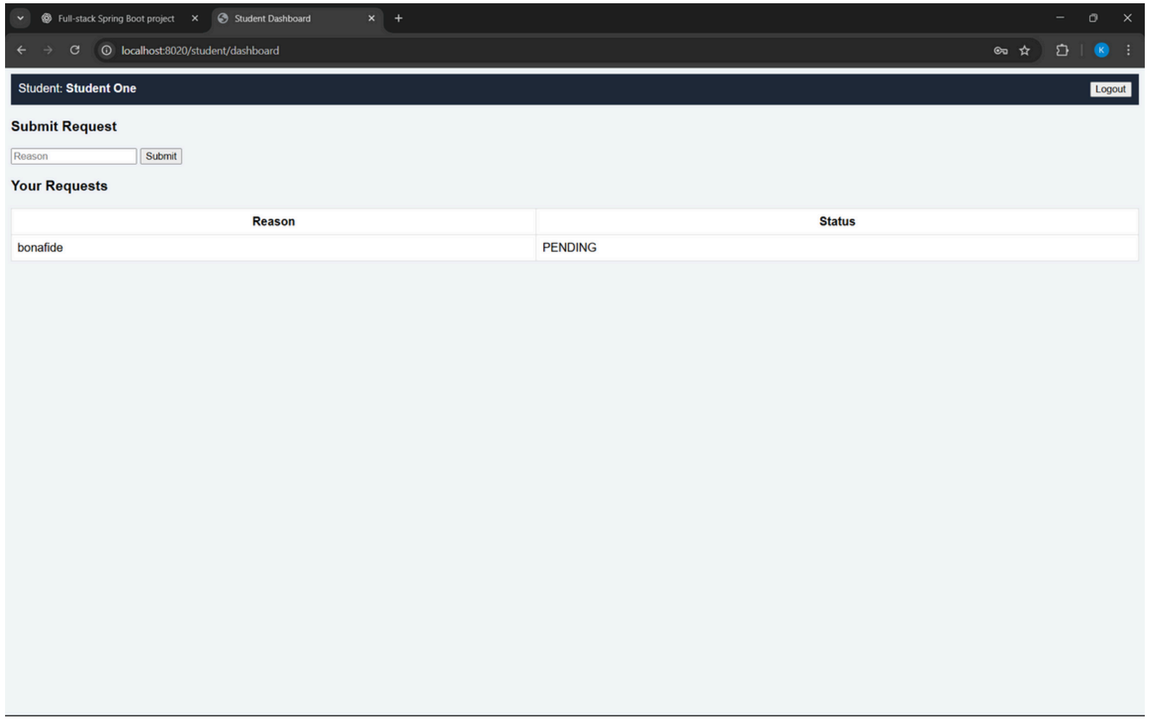
The Academic Request Management System successfully automates the process of handling academic requests in an institution. By implementing role-based access and structured workflows, the system ensures transparency, efficiency, and security. This project demonstrates the effective use of Spring Boot and database-driven web applications in solving real-world academic problems.

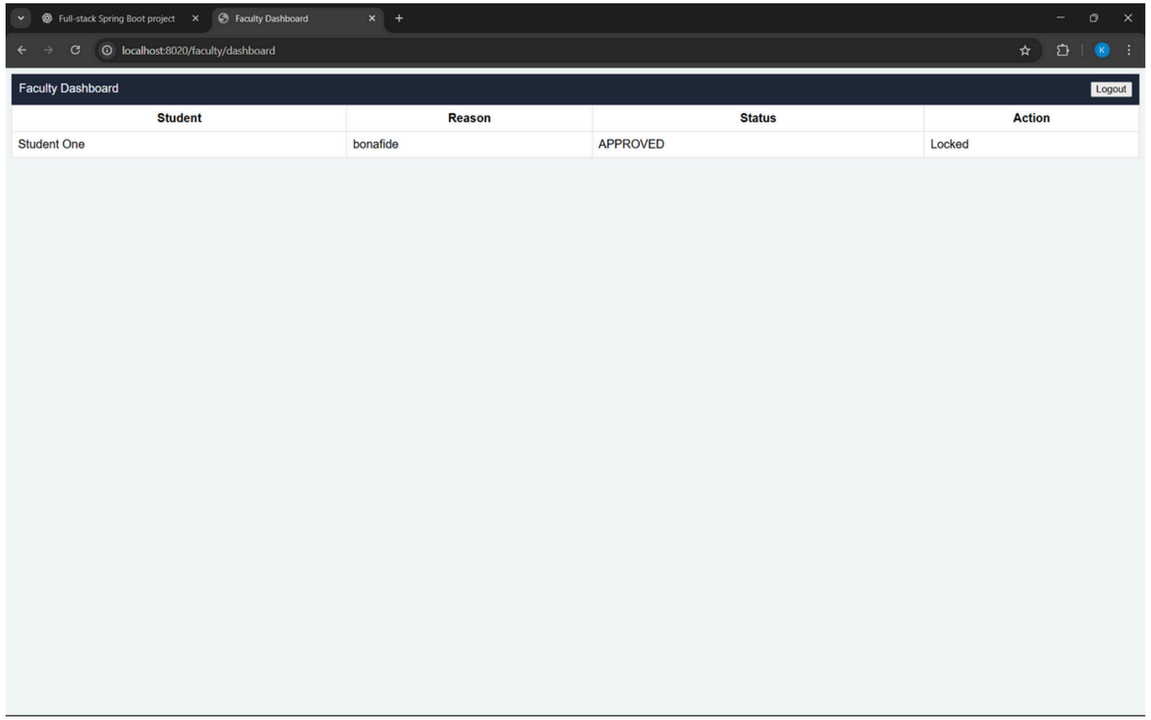
```
1  CREATE DATABASE careerpath_userdb;
2  SHOW DATABASES;
3  USE careerpath_userdb;
4  select * from users;
5
```

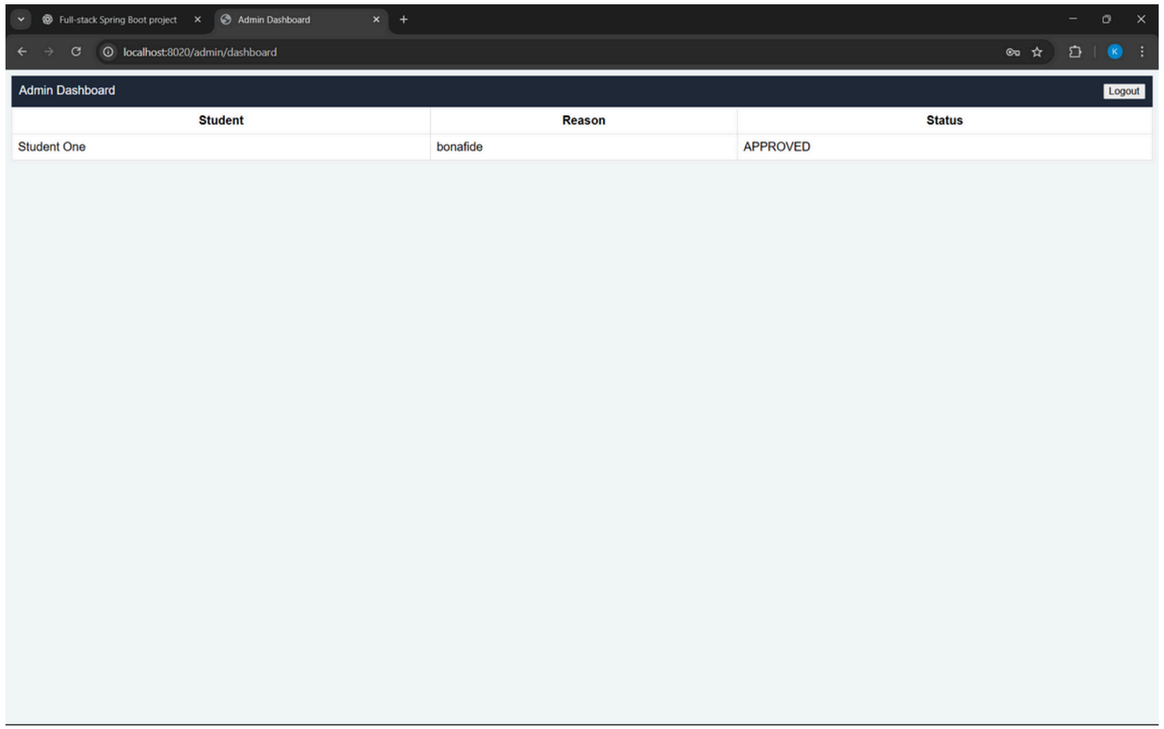
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	created_at	email	enabled	password	role
1	2025-12-17 14:09:53.375470	testuser@gmail.com	1	$2a$10$eIvlyIOPmg7jZcuq/nk9G.7nuMQoAIXq...	STUDENT
2	2025-12-18 16:59:12.500562	kamaleshs.saravanan@gmail.com	1	$2a$10$NwZ1tCzwUwCOkWCtXqisLu1ZbQru4Y...	STUDENT
3	2025-12-18 22:30:06.307845	student@gmail.com	1	$2a$10$q8R62VMFRWjghqddLK2Fh.RnzCYkybh...	STUDENT
NULL	NULL	NULL	NULL	NULL	NULL




Full-stack Spring Boot project
Student Dashboard
localhost:8020/student/dashboard
Student: Student One
Logout
Submit Request
Reason
Submit
Your Requests
Reason
Status
bonafide
PENDING


Full-stack Spring Boot project
Faculty Dashboard
localhost:8020/faculty/dashboard
Faculty Dashboard
Logout
Student
Reason
Status
Action
Student One
bonafide
APPROVED
Locked


Full-stack Spring Boot project
Admin Dashboard
localhost:8020/admin/dashboard
Admin Dashboard
Logout
Student
Reason
Status
Student One
bonafide
APPROVED