

Online Examination System

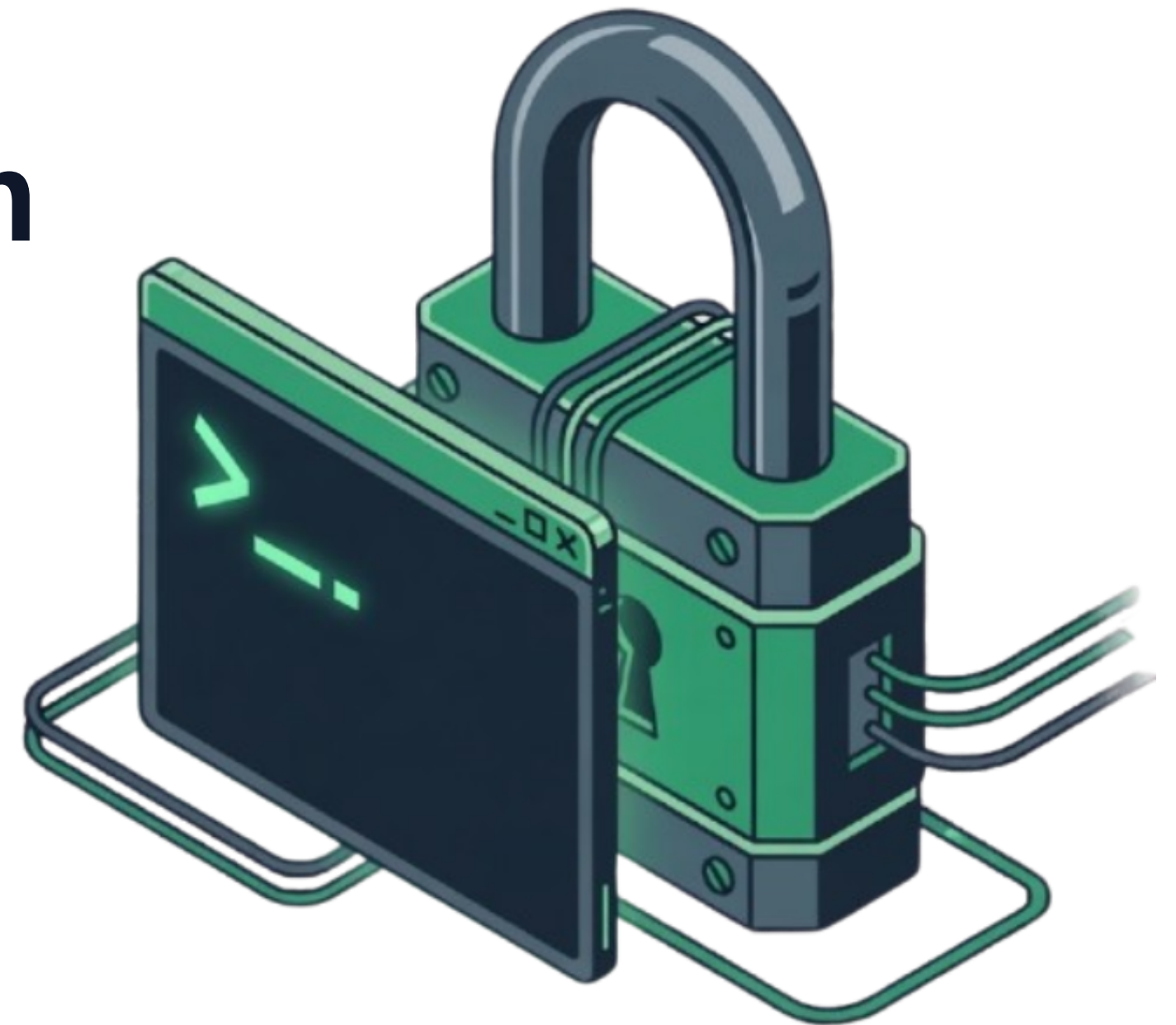
A Role-Based Web Platform for Secure and Efficient Online Examinations

Course: Object-Oriented Programming

Lecturer: SEK SOCHEAT

Student Name: [Sareth Pich] [Veasna PanhaReach] [Men Sereyyuth]

Institution: University of Technology and Entrepreneurship



Problem & Motivation

⚠ Traditional Limitations

- ✗ Manual examination preparation & distribution
- ✗ Time-consuming manual marking and score calculation
- ✗ Difficulty managing large volumes of questions securely
- ✗ High risk of unauthorized access to exam resources
- ✗ Lack of centralized records and data organization



💡 Proposed Motivation

- ✓ Centralized digital examination management
- ✓ Streamlined administrative and testing workflows
- ✓ Automated objective question evaluation
- ✓ Rigid Role-Based Access Control (RBAC): Student ≠ Teacher ≠ Admin
- ✓ Secure, structured relational database records

Transforming Examination Workflows

⚠ Traditional Limitations



- ✗ **Manual & Slow:** Preparation, distribution, and marking.
- ✗ **Poor Security:** Risk of unauthorized access and tampering.
- ✗ **Disorganized:** Lack of central records and data control.



- ✗ **Disorganized:** Lack of central records and data control.

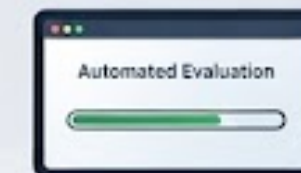


Digital
Transformation or
Evolution

💡 Proposed Motivation



- ✓ **Centralized & Fast:** Digital management and automated workflows.
- ✓ **Enhanced Security:** Secure relational database records.
- ✓ **Objective & Controlled:** Automated evaluation and role-based access.



Secure, Accessible, and Efficient Solution

Project Objectives

🎯 General Objective

Develop a secure and maintainable web-based Online Examination System using Python OOP and Flask that manages examinations through robust Role-Based Access Control (RBAC).

✓ Develop user authentication and granular authorization.

✓ Implement rigid Role-Based Access Control (RBAC).

✓ Allow teachers to create and manage examinations.

✓ Enable teachers to build dynamic question banks.

✓ Provide an intuitive online exam-taking interface.

✓ Automate grading for objective questions.

✓ Generate and securely store detailed results.

✓ Apply clean Python OOP and PostgreSQL principles.

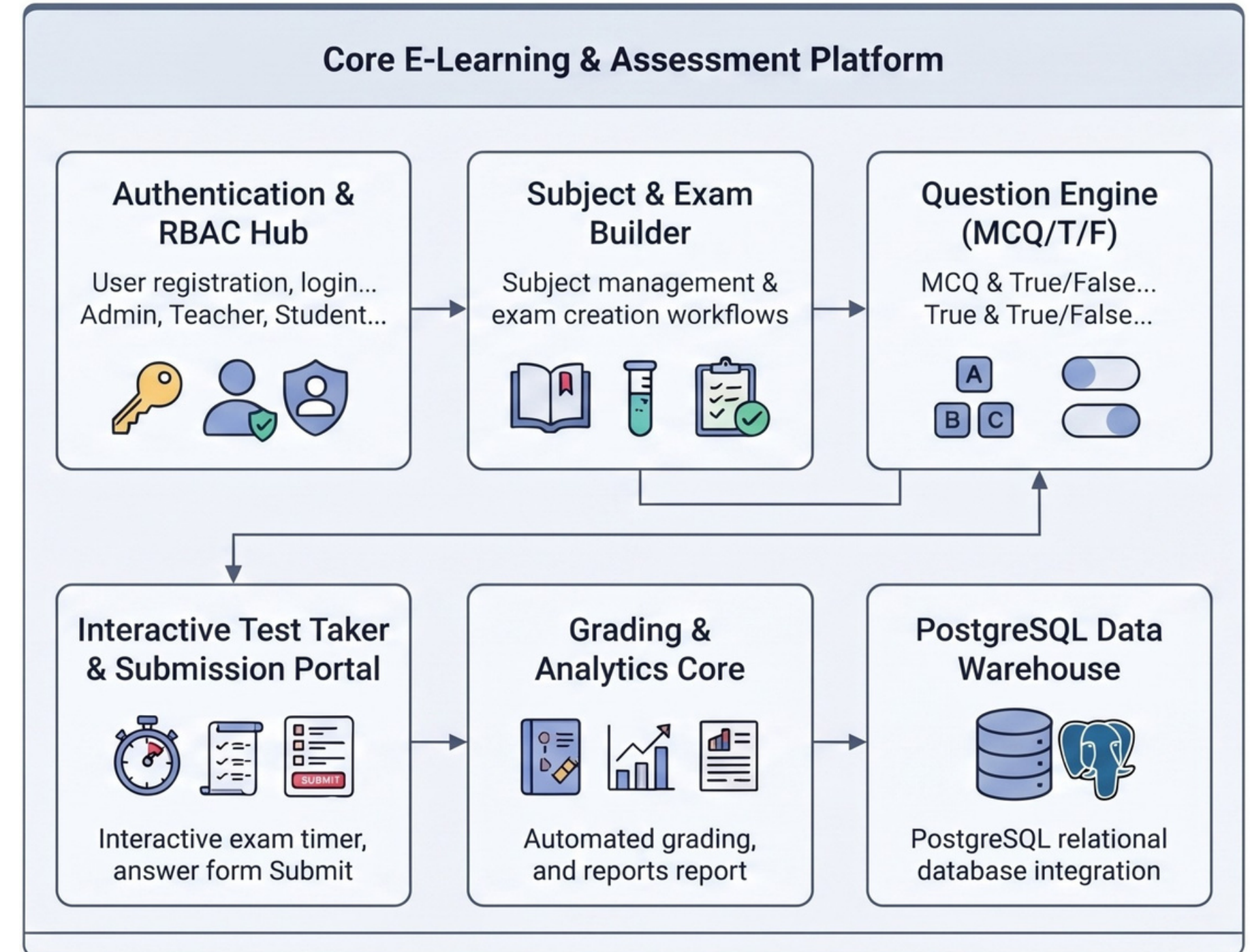
Project Scope

✓ In Scope (Current R&D)

- › User registration, login, and session control
- › Admin, Teacher, and Student user roles with RBAC
- › Subject management & exam creation workflows
- › MCQ & True/False question handling
- › Interactive exam timer & online answer submission
- › Automated grading and result history reporting
- › PostgreSQL relational database integration

🕒 Out of Scope / Future Scope

- Dedicated native mobile application
- Advanced AI-based essay grading algorithms
- Facial recognition & webcam proctoring
- Multi-university distributed cloud infrastructure



Target Users & User Stories

System Admin

Admin

Responsible for overall system maintenance and global configurations.

- Manage system users
- Assign roles & permissions
- Manage course subjects
- View system audit logs

Instructor

Teacher

Responsible for curriculum design, test authoring, and evaluation.

- Create examinations
- Manage question banks
- Set exam duration & rules
- Analyze student results

Examinee

Student

Responsible for taking assigned assessments and reviewing feedback.

- Secure system login
- View active examinations
- Take exams with timer
- Review historical results

Proposed Features & Module Priority

Priority 1 Core Features

Authentication: Secure login & sessions

RBAC: Strict permission enforcement

User Mgmt: Account administration

Exam Engine: Timed test delivery

Auto-Grading: Objective scoring

Priority 2 Supporting

Subject Management: Taxonomy setup

Question Bank: Reusable item pool

Exam History: Past performance logs

Scheduling: Active date windows

Randomization: Option shuffling

Priority 3 Future Enhancements

Advanced Analytics: Performance metrics

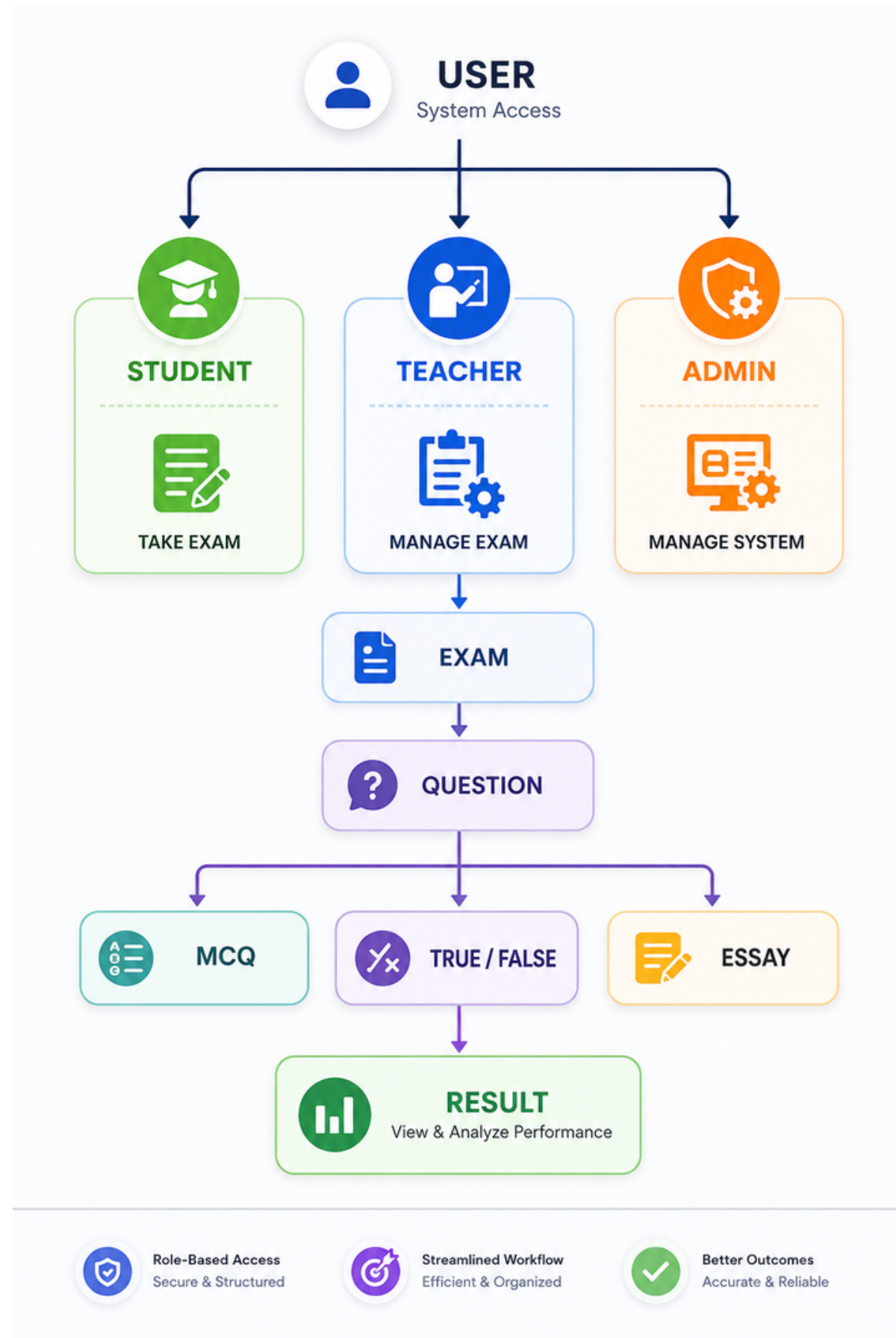
AI Essay Grading: NLP assistance

AI Question Gen: Automated drafting

Proctoring: Monitoring extensions

API/Mobile: Cross-platform support

Object-Oriented Design (OOP)



Encapsulation

Restricting direct data access in models (e.g., student answers, user credentials) using pythonic property getters and setters.

Inheritance

Specializing user types (`Student`, `Teacher`, `Admin`) and question formats from robust base classes.

Polymorphism

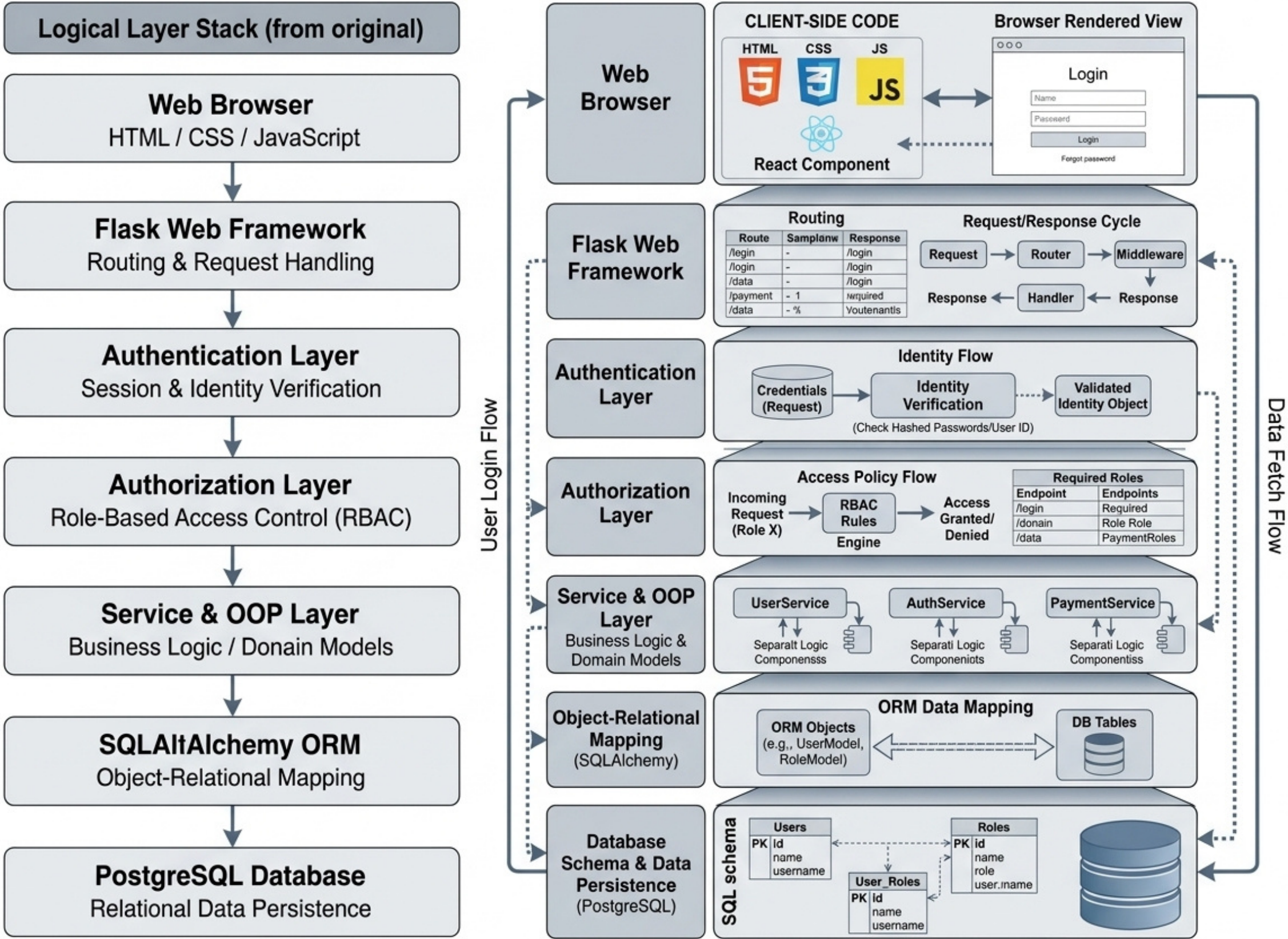
Unified evaluation interface where different question types execute custom grading logic.

Abstraction & Composition

Hiding low-level database details behind service classes; an Exam object cleanly composes multiple Question entities.

System Architecture

Integrated Layered Architecture and Data Flow Plan



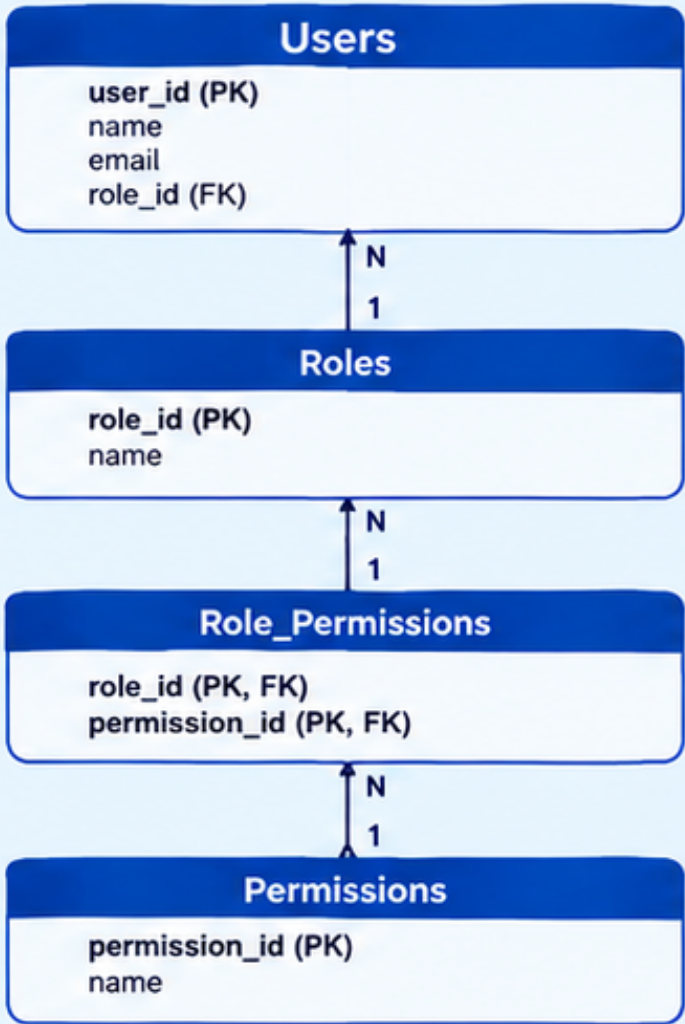
PostgreSQL Database Design

How the database connects security, examinations, and results



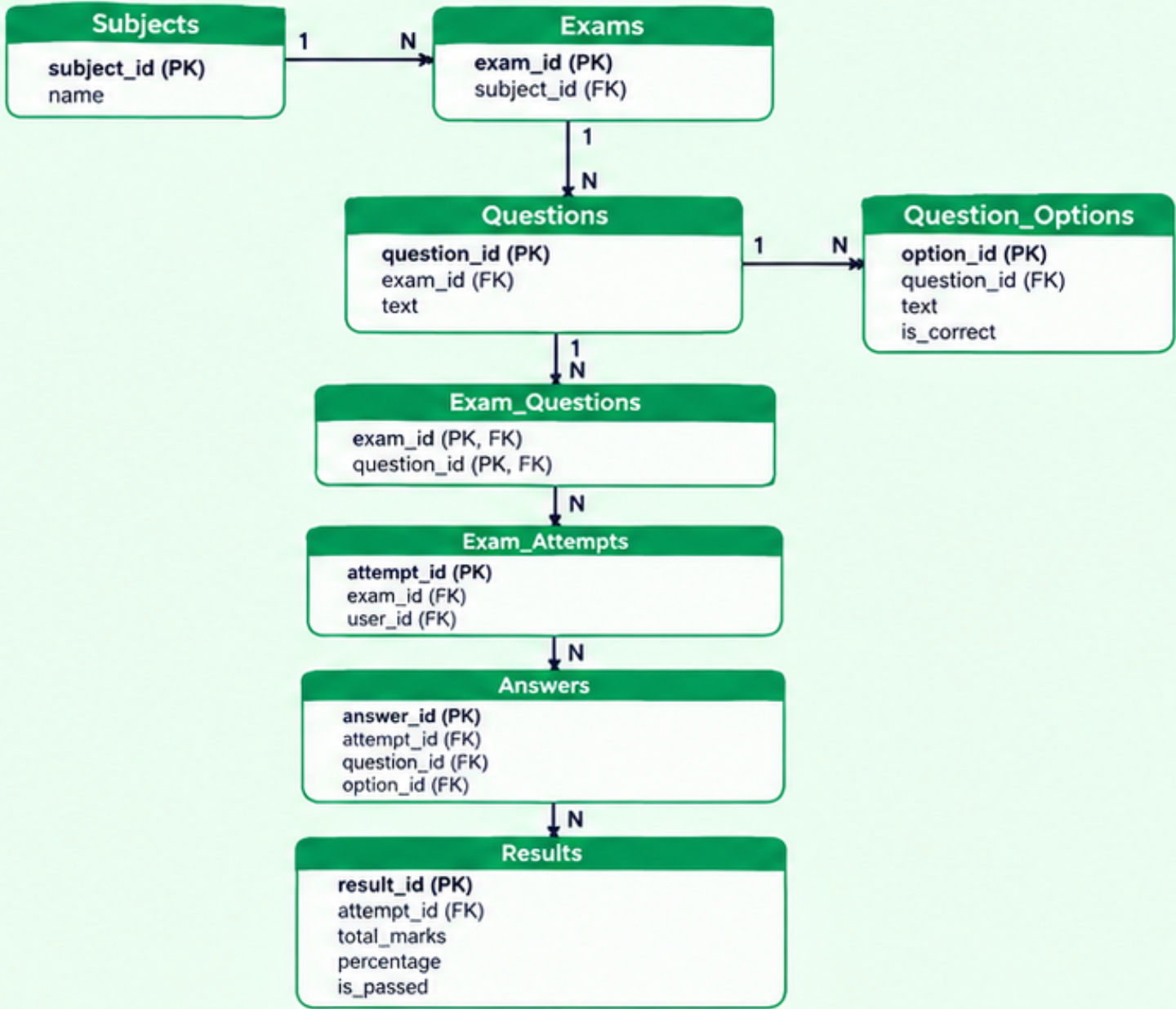
RBAC – Role-Based Access Control

Manage users, roles and permissions



Examination Management Flow

From subjects to results



Key Benefits



Secure & Scalable
RBAC for controlled access



Efficient Management
Easy exam creation & scheduling



Data Integrity
Accurate tracking of attempts & results

Development Methodology & Sprint Timeline

Four stages from foundation to a tested system



BUILD → SECURE → DEVELOP → TEST

Expected Project Deliverables

Three groups of outputs from the project

Application

- Python Flask web application
- PostgreSQL database
- Responsive user interface

Security

- Secure authentication
- Granular RBAC authorization
- Teacher & Admin management panels
- Automated objective evaluation

Documentation

- System architecture blueprints
- Detailed ERD documentation
- Object-oriented class diagrams
- Deployment & setup instructions

A secure • maintainable • documented examination platform

Q&A — Feedback & Technical Decisions

Three decisions we would like feedback on

RBAC

- Should Admin also manage exams?
- Or should exam management remain with Teachers?

OOP Design

- Are inheritance relationships optimal?
- Is our composition structure suitable?

Future Scope

- Should essay questions be in the first release?
- Or keep AI essay grading for later?

*Thank
you!*

WE WELCOME YOUR QUESTIONS & FEEDBACK