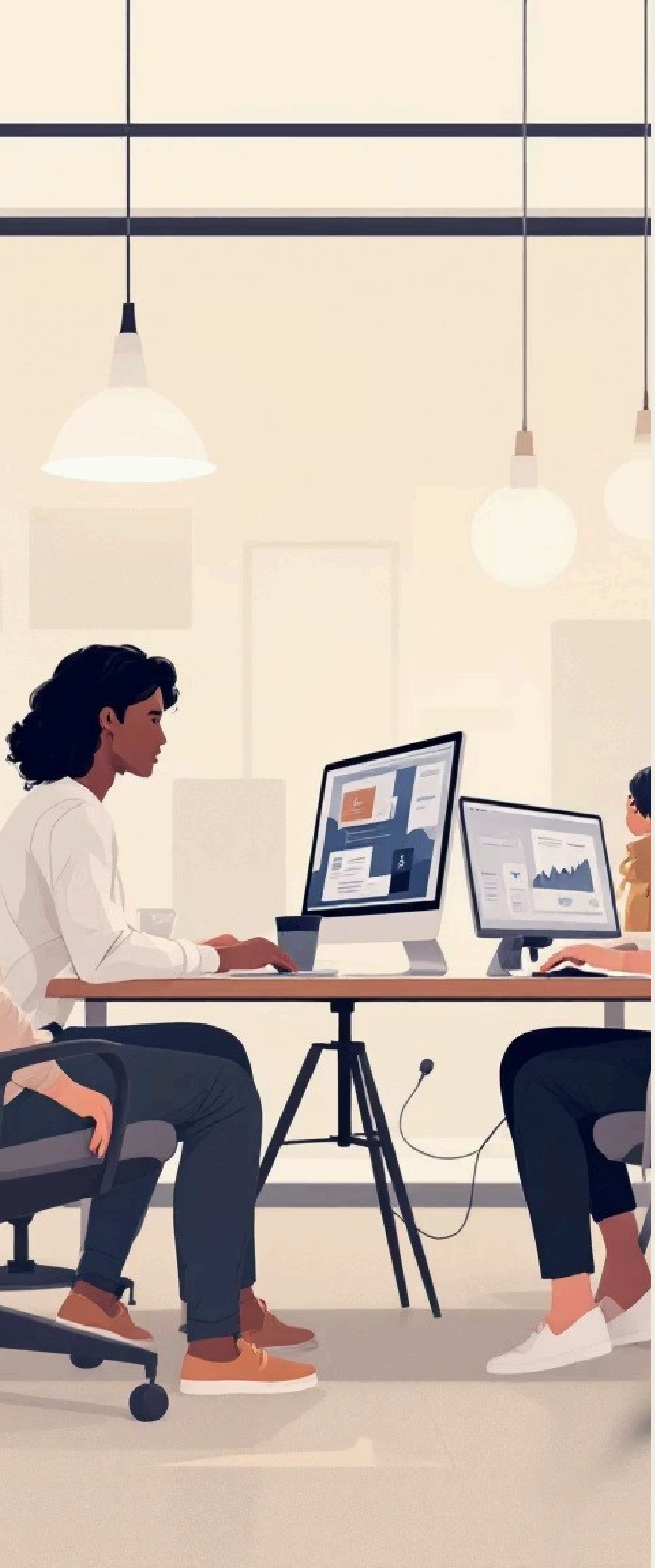
Efficient GUI-Based Attendance Management System

Keyfeatures of our AttendanceManagementSystem:

- Built with Java, JavaFX/Swing, OOP, JDBC, and MySQL.
- Modern graphical user interface with intuitive design.
- Manages students and records attendance visually.
- Generates comprehensive attendance reports with charts.
- Streamlines tracking for educational institutions.
- MVC architecture, secure database, and robust error handling.



Meet the Codes for Coders Team

1

Amandeep Singh Bhatia

24scse1011218

- Defined project structure
- Designed GUI interface using JavaFX/Swing
- Applied core Java & OOP

2

Yash Mishra

24scse1010914

- Developed attendance recording GUI
- Implemented user authentication screens
- Generated reports with visual charts

3

Yash Vardhan Singh Rana

24scse1010490

- Designed database schema
- Established JDBC connectivity
- Integrated backend with GUI

Project Overview

Demonstratessoftwaredevelopmentskillsusingindustry-standard technologies and GUI design patterns.

Core Java programming

- Collections
- Exception Handling
- I/O
- Event Handling

Implemented OOP principles

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

GUI Development

- JavaFX/Swing for interactive user interface

Reliable MySQL connectivity via JDBC API

Full CRUD operations for student & attendance records

MVC architecture for maintainability

- Model
- View
- Controller

Normalized MySQL relational database schema

Attmccre	Data	Tme	Werdaps	Ferct
Frot	23145	1E.00	14.00	0
TLse	25145	10	2	1
Srd4	15A4	29	22	6
○ 200R.00	21	1	0	5
Lare	20	0	0	1
Meor	20	7	7	0
	2	0	0	0

Key Features and Functionalities

Arobust Studentand Attendance ManagementSystem.

Built with Java, OOP, and MySQL, it efficiently manages student records and attendance using JDBC and a layered architecture.

Add New Student

- Register students via intuitive GUI forms
- Real-time input validation with visual feedback
- Auto-generates unique student IDs
- Stores records in MySQL database

View All Students

- Display students in interactive table/grid view
- Search and filter functionality
- Sort by ID, name, or course
- Click to view detailed student information

Mark Attendance

- Visual attendance marking interface
- Calendar widget for date selection
- Dropdown/radio buttons for Present/Absent
- Batch attendance marking capability

View Attendance History

- Interactive charts and graphs
- Filter by student, date range, or course
- Display attendance percentage with progress bars
- Export reports to PDF/Excel

Project Structure

TheprojectfollowsMVC(Model-View-Controller) architecture for GUI applications.

AttendanceManagementSystem/

01

Model Layer

- src/model/: Defines data structures and business objects
- Student.java: Student entity with attributes
- Attendance.java: Attendance record entity
- Encapsulates data with getter/setter methods

02

View Layer

- src/view/: Contains all GUI components
- LoginView.java: User authentication screen
- DashboardView.java: Main application window
- StudentView.java: Student management interface
- AttendanceView.java: Attendance marking screens
- ReportView.java: Report generation and visualization

03

Controller Layer

- src/controller/: Handles user interactions and business logic
- StudentController.java: Manages student operations
- AttendanceController.java: Handles attendance logic
- ReportController.java: Generates reports
- Coordinates between View and Model

04

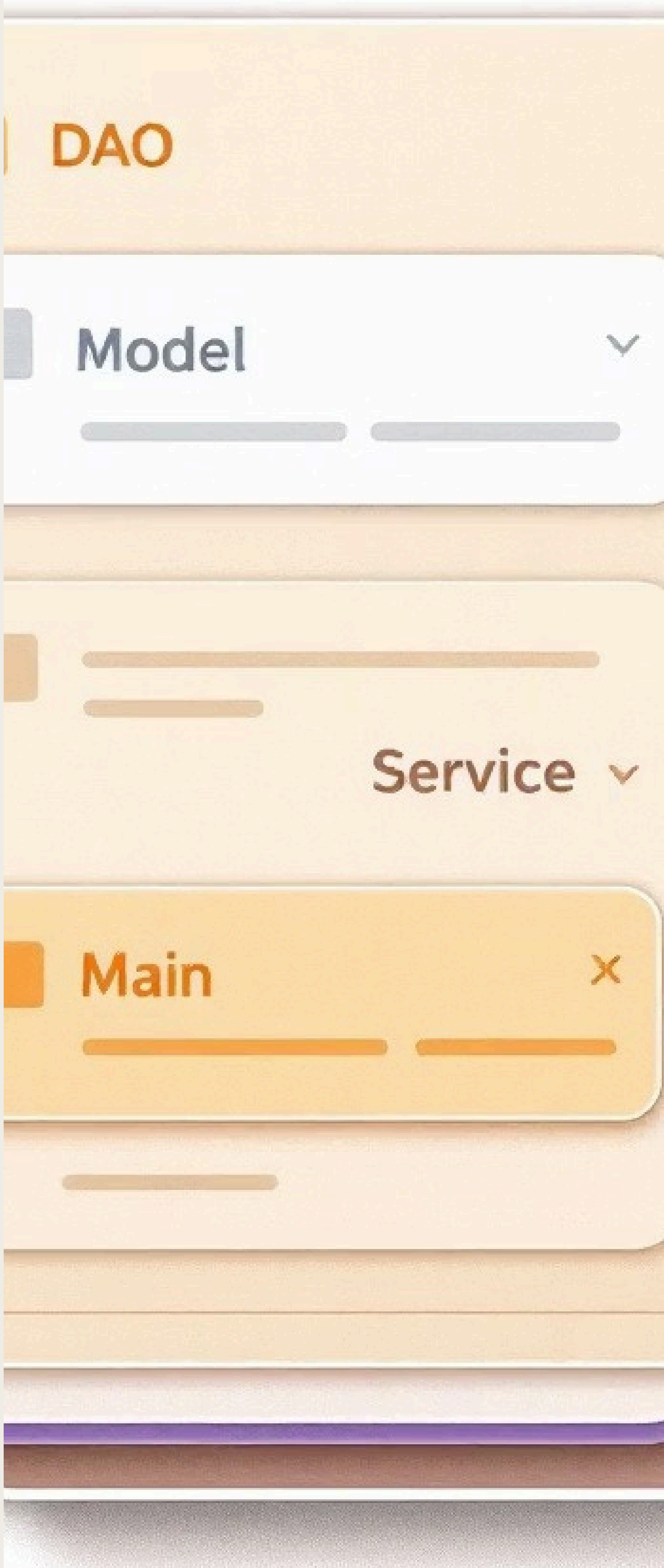
DAO Layer

- src/dao/: Database operations
- DBConnection.java: Connection management
- StudentDAO.java: Student CRUD operations
- AttendanceDAO.java: Attendance data operations

05

Resources & Configuration

- resources/db.properties: Database configuration
- resources/styles.css: GUI styling (for JavaFX)
- resources/images/: Application icons and images
- attendance.sql: Database schema
- README.md: Documentation



Database Schema (MySQL)

Normalized relational design with two interconnected tables for data integrity.

Database: **attendance_db**

Students Table (Primary):

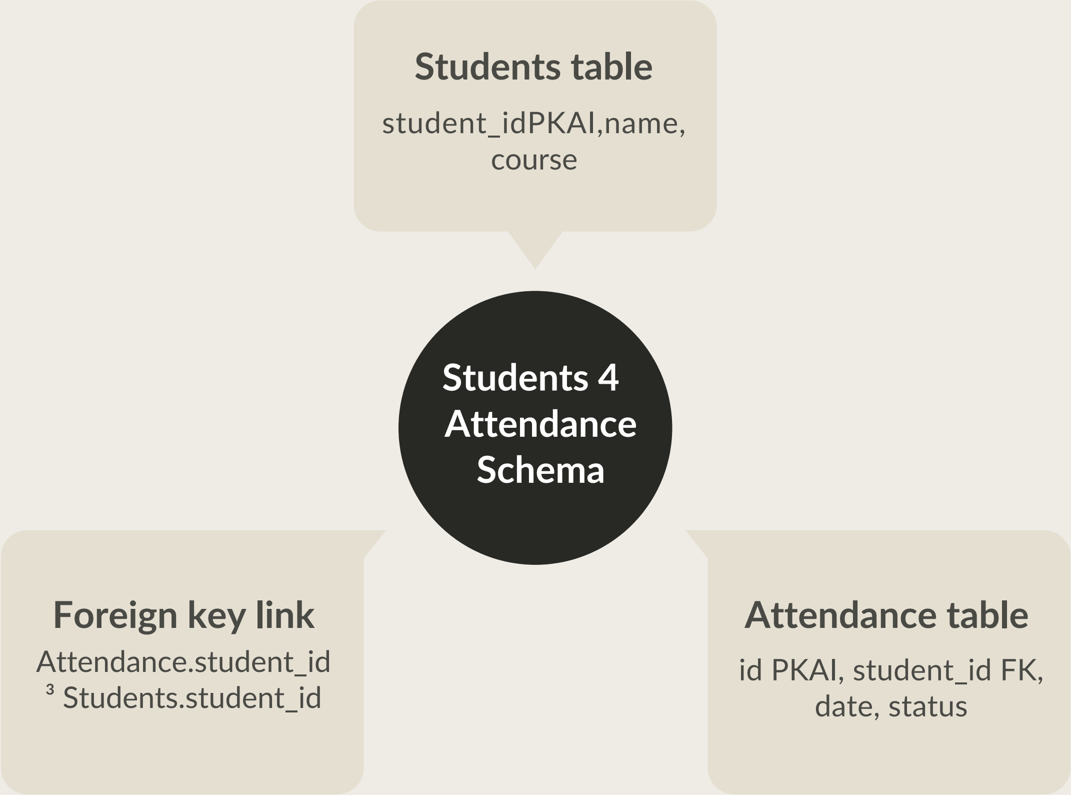
- student_id (PK, AI)
- name (VARCHAR)
- course (VARCHAR)
- Core student information

Attendance Table (Transaction):

- id (PK, AI)
- student_id (FK)
- date (DATE)
- status (VARCHAR)
- Daily attendance records

Relationship:

- Attendance.student_id links toStudents.student_id
- Ensures valid student attendance
- Maintains referential integrity
- Facilitates efficient JOIN queries



Students table

student_idPKAI,name, course

Students 4 Attendance Schema

Foreign key link

Attendance.student_id ^{3} Students.student_id

Attendance table

id PKAI, student_id FK, date, status

JDBC Configuration

Databaseconnection is configuredusingapropertiesfile for secure credential managementandeasy updates.

`db.properties` File:

```
db.url = jdbc:mysql://localhost:3306/attendance_db
db.username = root
db.password = your_password
db.driver = com.mysql.cj.jdbc.Driver
```

Configuration Details:

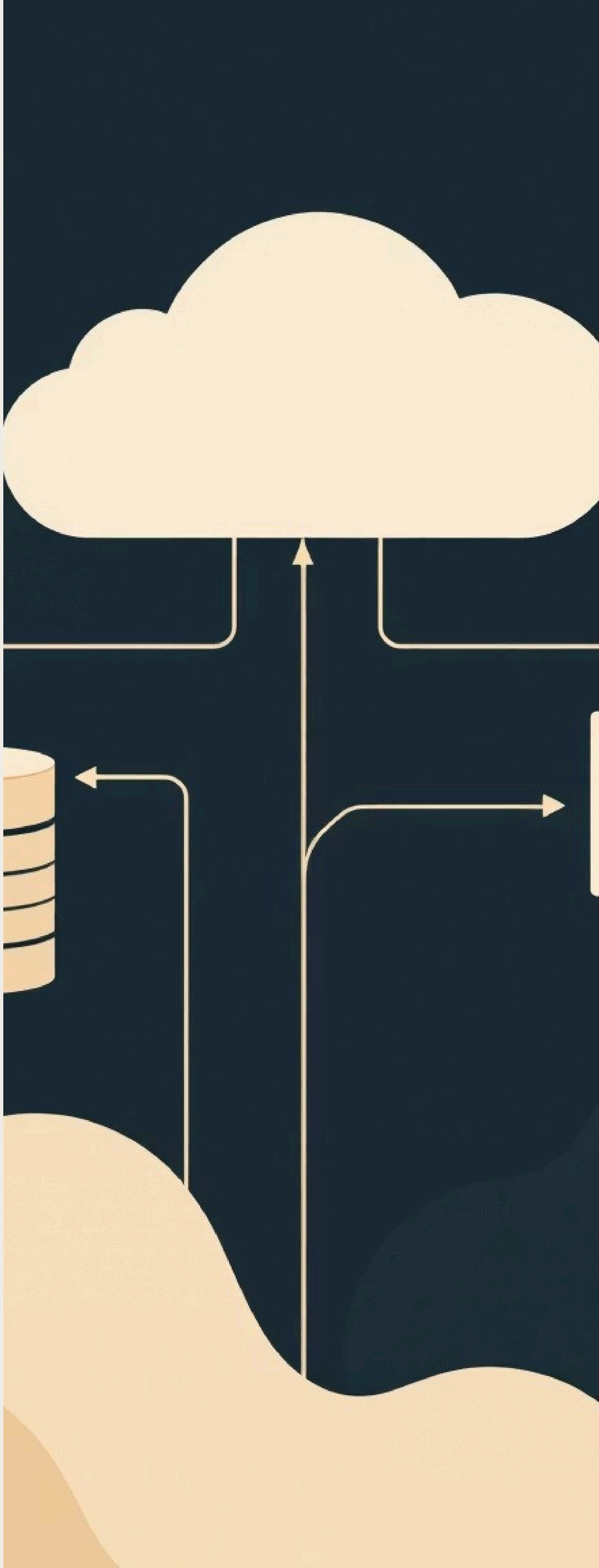
- `db.url`: JDBC connection string.
- `db.username`: Database user.
- `db.password`: Secure password.
- `db.driver`: MySQL Connector/J driver.

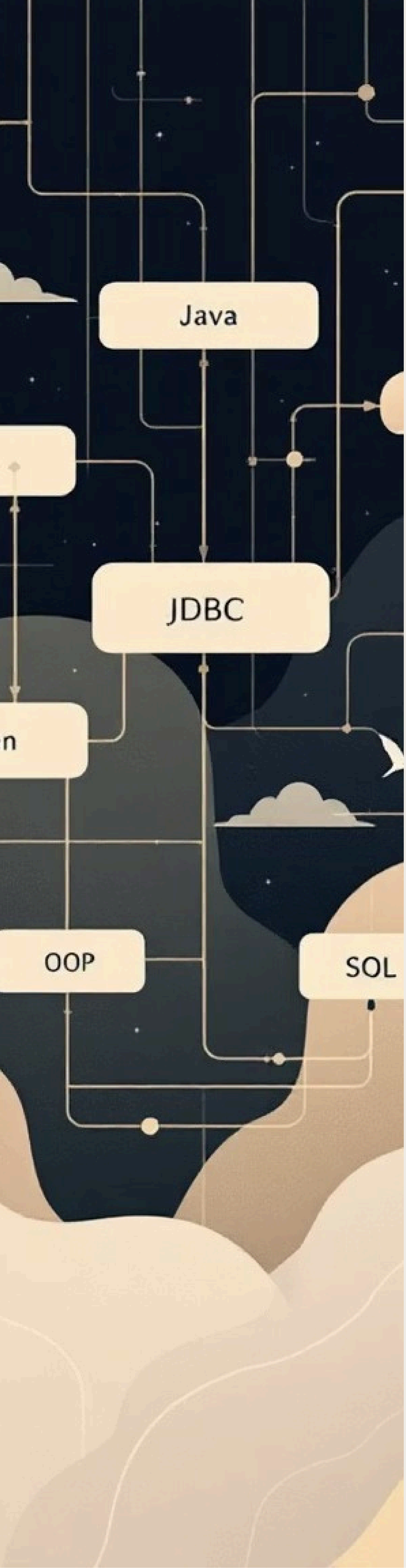
Implementation Details(`DBConnection.java`):

- Reads properties (`Properties` class).
- Establishes connection (`DriverManager`).
- Uses connection pooling.
- Handles `SQLException` (error logging).
- Ensures connection closure.
- Simplifies environment migration.

Benefits:

- Separates config from code.
- Allows credential changes without recompile.
- Externalized configuration.
- Simplifies deployment.





Technologies Used

Builtusingindustry-standardtechnologiesfordesktop GUI applications.

Java: Core programming language

- Version: Java SE 8 or higher
- Collections, exception handling, event handling
- Platform-independent execution

JavaFX / Swing: GUI framework

- Rich UI components (tables, forms, charts)
- Event-driven programming model
- CSS styling support (JavaFX)

JDBC: Java Database Connectivity

- For MySQL
- PreparedStatement for security
- Connection pooling and transactions

MySQL: Relational database

- Version: MySQL 8.0 or compatible
- ACID compliance for data integrity

Maven: Dependency management

- Manages JavaFX/Swing and MySQL libraries
- Standardized project structure

MVC Pattern: Architecture

- Separates UI from business logic
- Improves maintainability and testability

Scene Builder: Visual layout tool

- Drag-and-drop UI design
- FXML file generation

How to Run the Project

Step-by-stepguidetoruntheGUI-basedAttendanceManagement System.

01

1. Clone the Repository

```
git clone
https://github.com/yourusername/AttendanceManagementSystem.git
cd AttendanceManagementSystem
```

- Ensure Git is installed.

02

2. Configure the Database

- Install MySQL Server (version 8.0+).
- Run: `mysql -u root -p attendance_db < attendance.sql`
- Update `db.properties` with your credentials.
- Verify tables are created.

03

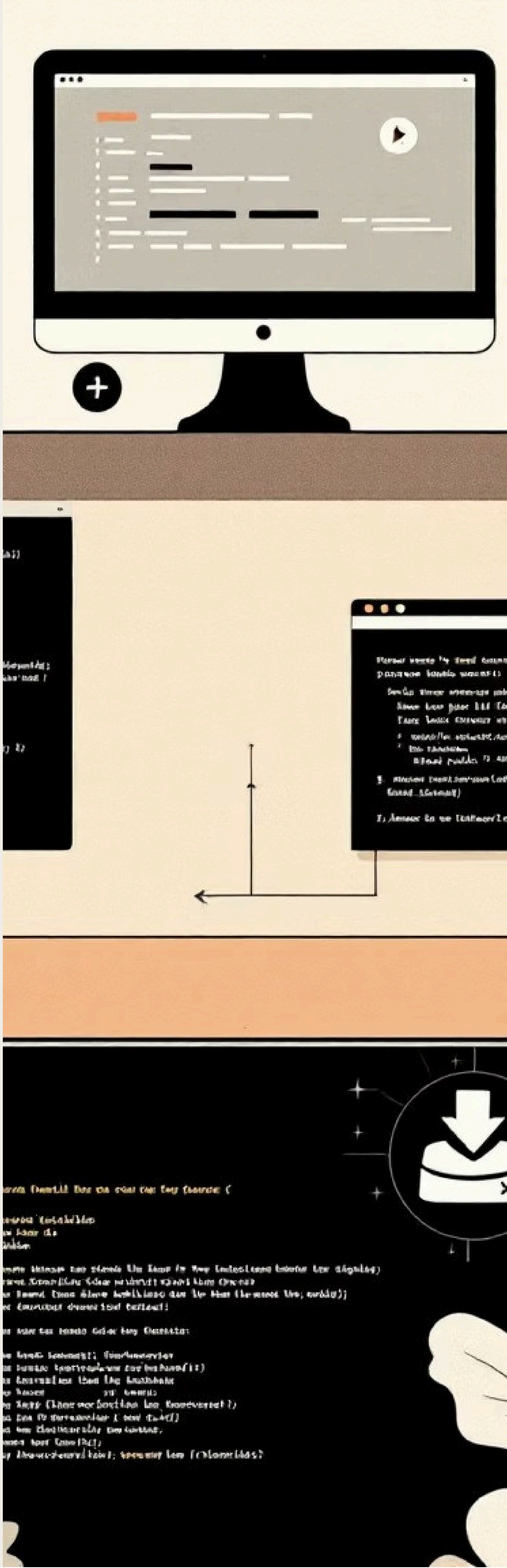
3. Add Required Libraries

- If using Maven: Dependencies auto-download (JavaFX/Swing, MySQL Connector).
- If not using Maven:
    - Download JARs manually.
    - JavaFX SDK or Swing (included in JDK).
    - MySQL Connector/J (version 8.0.33).
- Add JARs to project classpath.

04

4. Run theApplication

- For JavaFX: `java--module-path/path/to/javafx-sdk/lib --add-modules javafx.controls,javafx.fxml-jarAttendanceSystem.jar`
- For Swing: `java-jarAttendanceSystem.jar`
- Or run from IDE (IntelliJ, Eclipse, NetBeans).
- Login screen will appear.
- Default credentials: admin/admin.



GUI Interface Overview

Thegraphicaluserinterfaceprovidesanintuitiveandmodernexperience for managingattendance.

Main Interface Components:

Login Screen:

- Username and password fields
- Login button with validation
- Error messages for invalid credentials
- Clean, professional design

Dashboard:

- Navigation menu (Students, Attendance, Reports)
- Quick statistics cards (Total Students, Today's Attendance, etc.)
- Recent activity feed
- Logout button

Student Management:

- Table view with all students
- Add/Edit/Delete buttons
- Search bar with real-time filtering
- Form dialogs for data entry

Attendance Marking:

- Date picker for selecting date
- Student list with checkboxes/radio buttons
- Mark All Present/Absent buttons
- Save and Cancel options
- Confirmation dialogs

Reports & Analytics:

- Attendance charts (bar, pie, line graphs)
- Date range selector
- Export to PDF/Excel buttons
- Print preview functionality

Interface Features:

Responsive Layout

- Adapting to window size
- Flexible design
- Optimized for various screens

Keyboard Shortcuts

- For common actions
- Enhanced user efficiency
- Intuitive key assignments

Tooltips & Guidance

- Contextual help for users
- On-hover explanations
- Reduces learning curve

Visual Feedback

- Success/error messages
- Interactive elements
- Clear state indicators

Modern Design

- Contemporary color scheme
- Consistent icons Aesthetic
- appeal User-friendly visuals
-

GUI Screenshots & Features

Show the main screens of the application:

Login Screen:

- Clean login form with username/password
- Remember me checkbox
- Login button with validation

Dashboard:

- Navigation sidebar with menu items
- Statistics cards showing key metrics
- Recent activity section Quick action
- buttons

Student Management:

- Data table with student records
- Add/Edit/Delete action buttons
- Search and filter controls
- Pagination for large datasets

Attendance Interface:

- Calendar for date selection
- Student list with attendance status
- Mark Present/Absent controls
- Save and submit buttons

Reports View:

- Visual charts (bar, pie, line)
- Date range filters
- Export options (PDF, Excel)
- Print preview

System Benefits & Advantages

The GUI-based system offers enhanced usability and modern features.

User-Friendly Interface

- Intuitive graphical interface
- No command-line knowledge required
- Visual feedback for all actions
- Reduces training time for users

MVC Architecture

- Clear separation: Model, View, Controller
- Easy to maintain and extend
- Independent testing of components
- Supports multiple views for same data

Rich Visual Experience

- Interactive tables and forms
- Charts and graphs for analytics
- Color-coded status indicators
- Professional appearance

Enhanced Functionality

- Real-time data validation
- Batch operations (mark multiple attendance)
- Advanced search and filtering
- Export reports to multiple formats

Cross-Platform Compatibility

- Runs on Windows, Mac, Linux
- Consistent look and feel
- No browser required (desktop app)

Easy to Extend

- Add new screens and features
- Integrate with other systems
- Support for plugins/modules
- Can add web interface later

Secure and Robust

- User authentication with roles
- Input validation prevents errors
- PreparedStatement prevents SQL injection
- Exception handling with user-friendly messages

Scalable Design

- Handles growing data efficiently
- Database design supports expansion
- Ready for cloud deployment
- Can add mobile app integration

Project Conclusion

TheGUI-basedAttendanceManagementSystemdemonstrates modern desktop application development with user-centric design.

Successful System Development

- Developed robust GUI-based system
- All features implemented with visual interface
- Thoroughly tested for usability and reliability
- Professional-grade desktop application

Applied Core Technologies

- Java SE with JavaFX/Swing for GUI
- Implemented OOP and MVC patterns
- JDBC for database connectivity
- MySQL for data management
- Event-driven programming

Modern Architecture

- MVC pattern for clean separation
- View layer independent of business logic
- Reusable controller components
- Follows GUI design best practices

User-Centric Solution

- Intuitive interface for non-technical users
- Reduces manual errors with validation
- Visual feedback improves user experience
- Suitable for educational institutions

Valuable Learning Experience

- GUI development with JavaFX/Swing
- Event handling and user interaction
- Database integration with GUI
- MVC architecture implementation
- User experience design principles

Scalable Foundation

- Ready for additional features (notifications, email)
- Can integrate with web services
- Supports role-based access control
- Prepared for cloud deployment
- Can extend to mobile platforms
- Foundation for enterprise-level system