

Codelingo- GUI BASED ONLINE LANGUAGE LEARNING PLATFORM

TEAM MEMBERS:

Aakashi Chaudhary (24SCSE1280036)

Vaishnavi Yadav (24SCSE1280041)

Anushka Yadav (24scse1280003)

TECHNOLOGIES USED:

Swing, JDBC, MySQL, OOP, Collections,
Threading



OVERVIEW

This project, CodeLingo, is a GUI-based Online Language Learning Platform developed in Java using Swing for the interface and MySQL for backend data storage through JDBC. It follows Object-Oriented Programming principles such as inheritance, polymorphism, abstraction, and interfaces while also implementing collections and multithreading for smooth performance. The system includes three main modules—Admin, Instructor, and Learner—each with dedicated functionalities like user management, lesson creation, and progress tracking. DAO classes handle database operations cleanly, ensuring secure and efficient interaction with the MySQL tables. Overall, the project demonstrates complete integration of GUI, database connectivity, OOP concepts, and user-centric features in one cohesive application.



PROBLEM

Language learning lacks a simple system for managing lessons and tracking progress. This project provides a Java-based GUI platform with MySQL support to help admins manage users, instructors create lessons, and learners track their learning easily.





OBJECTIVES

- To provide a simple GUI platform for learning languages.
- To enable admins to manage users easily.
- To help instructors create and organize lessons.
- To allow learners to track their progress effectively.
- To integrate Java Swing with MySQL using JDBC for smooth data handling.

SYSTEM ARCHITECTURE

The system follows a simple three-layer architecture:

- GUI Layer using Java Swing for all user interfaces.
- Service + OOP Layer for handling logic, user roles, and interactions.
- Database Layer using JDBC to connect the application to a MySQL database for storing users, lessons, and progress.
- These layers work together to provide smooth communication between the interface, application logic, and backend data.





★ OOPS CONCEPTS USED: ★

The project applies key OOP concepts:

- Inheritance: Admin, Instructor, and Learner extend a common User class.
- Polymorphism: Each user type shows its own dashboard behavior.
- Abstraction: Shared user features are defined in an abstract class.
- Interfaces: ProgressTrackable interface is implemented by Learner.
- Encapsulation: Data such as user details and progress are secured through getters and setters.

JDBC AND DATABASE DESIGN

The project uses JDBC to connect Java Swing with a MySQL database. All data—users, lessons, and progress—is stored in structured tables, and DAO classes handle the database operations.



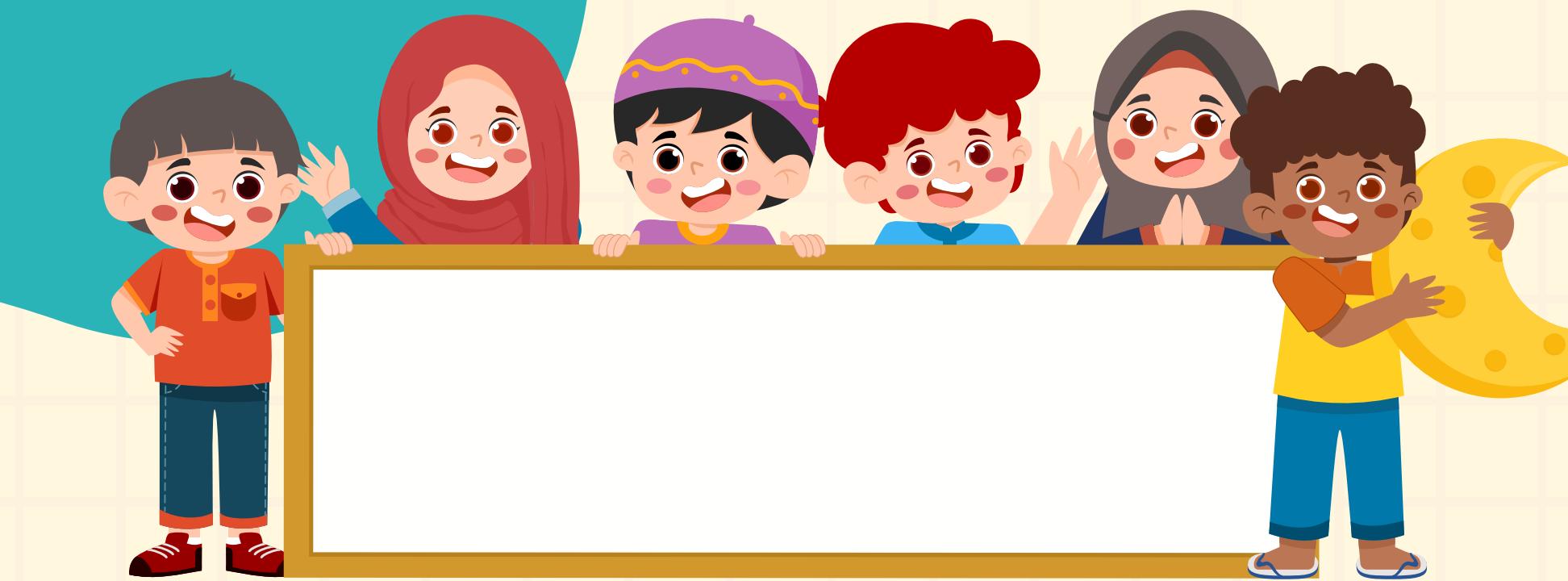
SERVLET INTEGRATION

- A Java Servlet is implemented to demonstrate server-side processing.
- The servlet handles login validation using JDBC.
- It connects to the same MySQL database used by the Swing GUI.
- This ensures proper separation between client and server logic.



CLIENT–SERVER ARCHITECTURE

- Java Swing application acts as the client interface.
- Java Servlet works as the server-side component.
- MySQL database is shared by both GUI and Servlet.
- JDBC enables secure communication with the database.



VALIDATION & ERROR HANDLING

- ★ Input validation implemented for login and lesson creation.
- ★ try-catch blocks handle database and runtime exceptions.
- ★ Invalid inputs are prevented from crashing the system.
- ★ User-friendly error messages are displayed through the GUI.



EVENT HANDLING & PROCESSING

- ActionListeners handle button clicks and user actions.
- ChangeListeners manage progress tracking slider events.
- Efficient event delegation improves application performance.
- Background thread manages auto-save functionality.



TESTING & EXECUTION

- All user roles tested: Admin, Instructor, and Learner.
- Database operations verified using MySQL.
- Servlet deployed and tested on Apache Tomcat.
- Input validation and error handling tested for robustness.





FUNCTIONAL MODULES:

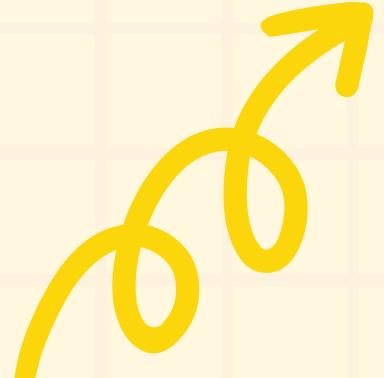
- Admin: Manages users and oversees system activity.
- Instructor: Creates lessons and supports learners.
- Learner: Views lessons and tracks learning progress.



KEY FEATURES:

- User roles with separate dashboards (Admin, Instructor, Learner).
- GUI created using Java Swing for easy interaction.
- Lesson creation and management by instructors.
- Learner progress tracking with auto-save.
- MySQL database integration using JDBC.
- OOP concepts, collections, and multithreading applied throughout.





THE PROJECT INTEGRATES A JAVA SWING GUI WITH JDBC AND A JAVA SERVLET TO DEMONSTRATE CLIENT–SERVER INTERACTION. IT ENSURES ROBUST VALIDATION AND A CLEAN, MODULAR DESIGN WHILE EFFECTIVELY SHOWCASING SERVER-SIDE PROCESSING AND SYSTEM INTEGRATION.

