CODELINGO- GUI BASED ONLINE LANGUAGE LEARNING PLATFORM

TEAM MEMBERS:

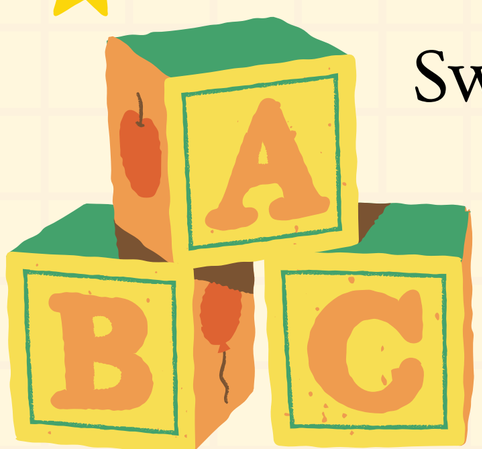
Aakashi Chaudhary (24SCSE1280036)

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.

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.

THIS PROJECT SUCCESSFULLY DEMONSTRATES A FUNCTIONAL LANGUAGE LEARNING PLATFORM USING JAVA SWING AND MYSQL. IT APPLIES KEY CONCEPTS LIKE OOP, JDBC, COLLECTIONS, AND MULTITHREADING TO BUILD A CLEAN, ROLE-BASED SYSTEM FOR ADMINS, INSTRUCTORS, AND LEARNERS. OVERALL, IT PROVIDES AN EFFECTIVE AND USER-FRIENDLY SOLUTION FOR MANAGING LESSONS AND TRACKING LEARNING PROGRESS.