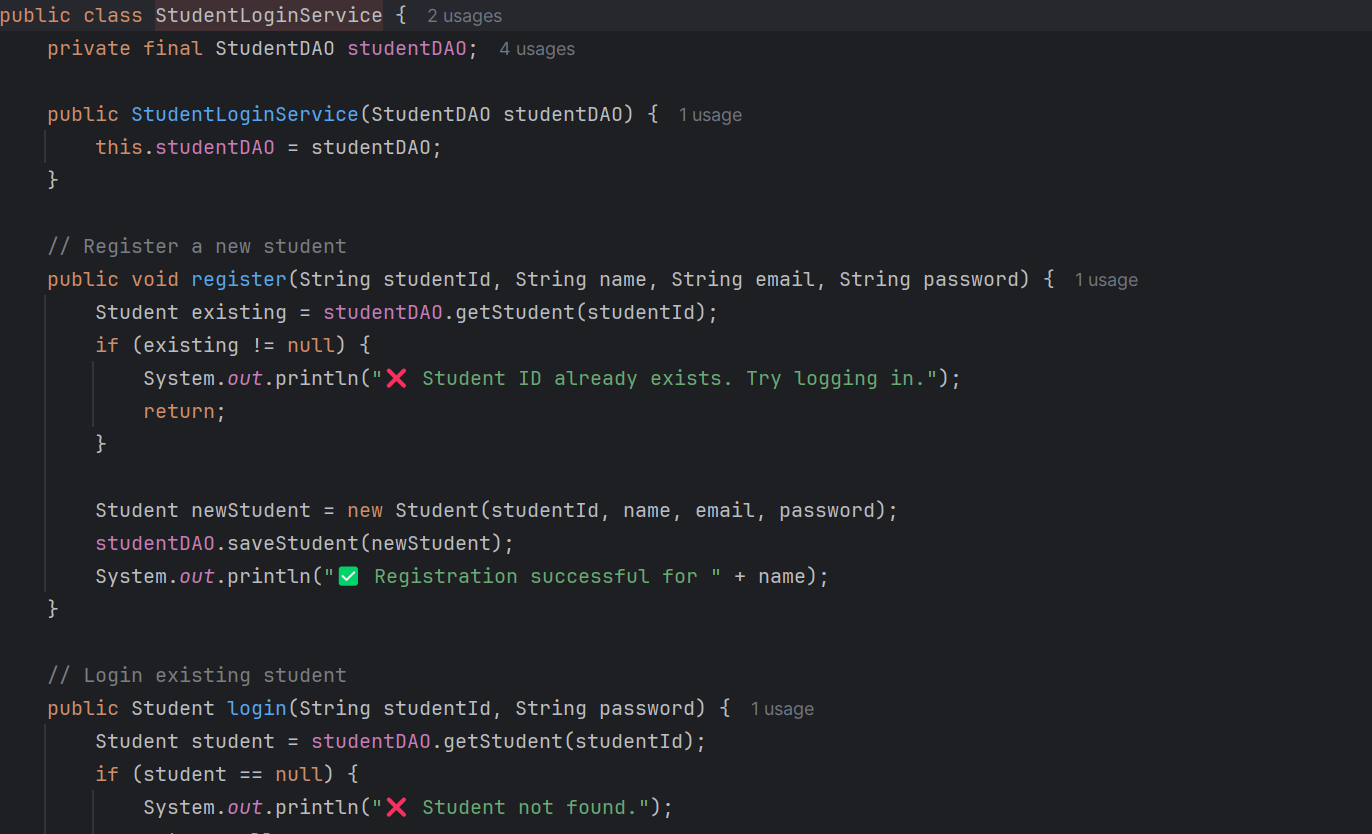
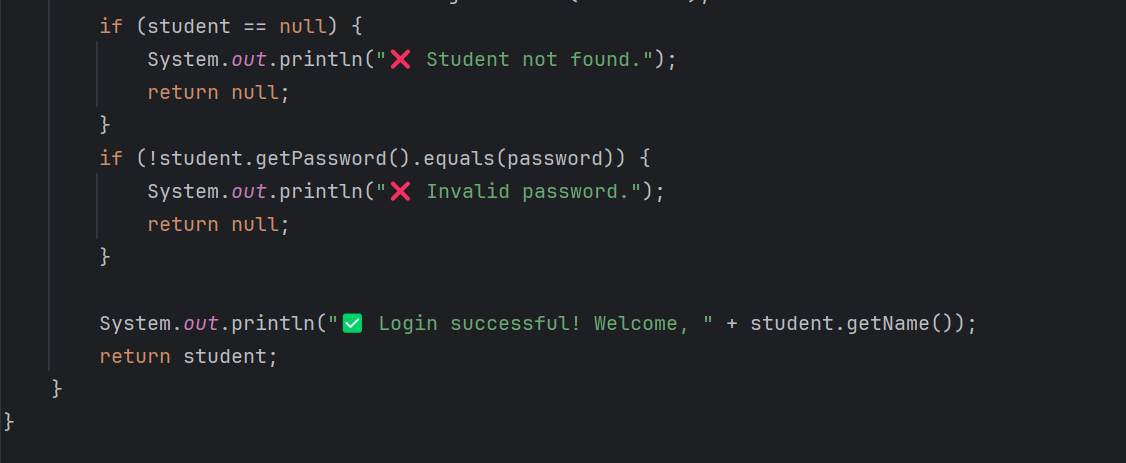
**Day 7 - DynamoDB- Student logs in NoSQL.**

**StudentLoginService.java**

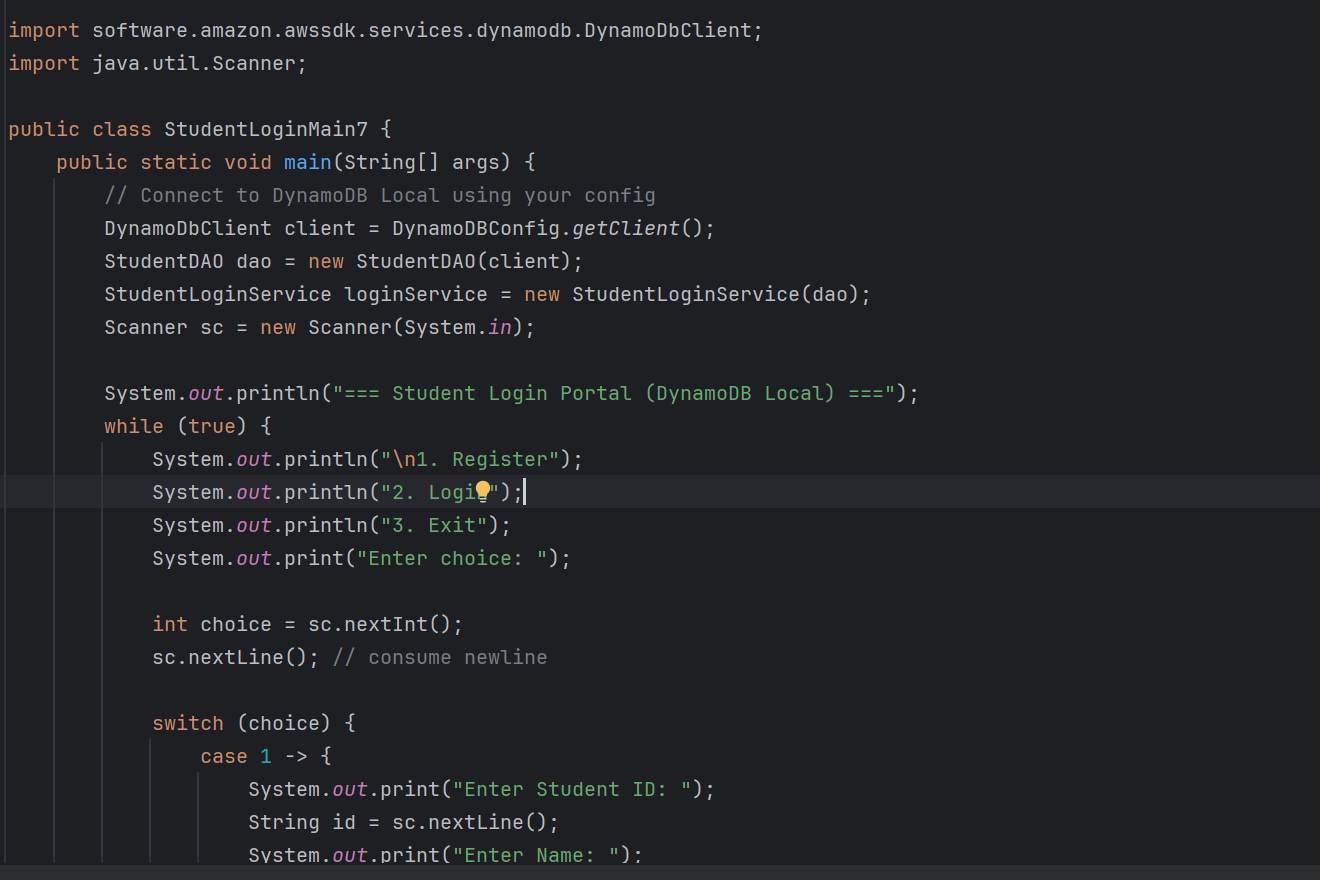
package PROJECT\_SCRS;  
  
public class StudentLoginService {  
 private final StudentDAO studentDAO;  
  
 public StudentLoginService(StudentDAO studentDAO) {  
 this.studentDAO = studentDAO;  
 }  
  
 // Register a new student  
 public void register(String studentId, String name, String email, String password) {  
 Student existing = studentDAO.getStudent(studentId);  
 if (existing != null) {  
 System.*out*.println("❌ Student ID already exists. Try logging in.");  
 return;  
 }  
  
 Student newStudent = new Student(studentId, name, email, password);  
 studentDAO.saveStudent(newStudent);  
 System.*out*.println("✅ Registration successful for " + name);  
 }  
  
 // Login existing student  
 public Student login(String studentId, String password) {  
 Student student = studentDAO.getStudent(studentId);  
 if (student == null) {  
 System.*out*.println("❌ Student not found.");  
 return null;  
 }  
 if (!student.getPassword().equals(password)) {  
 System.*out*.println("❌ Invalid password.");  
 return null;  
 }  
  
 System.*out*.println("✅ Login successful! Welcome, " + student.getName());  
 return student;  
 }  
}

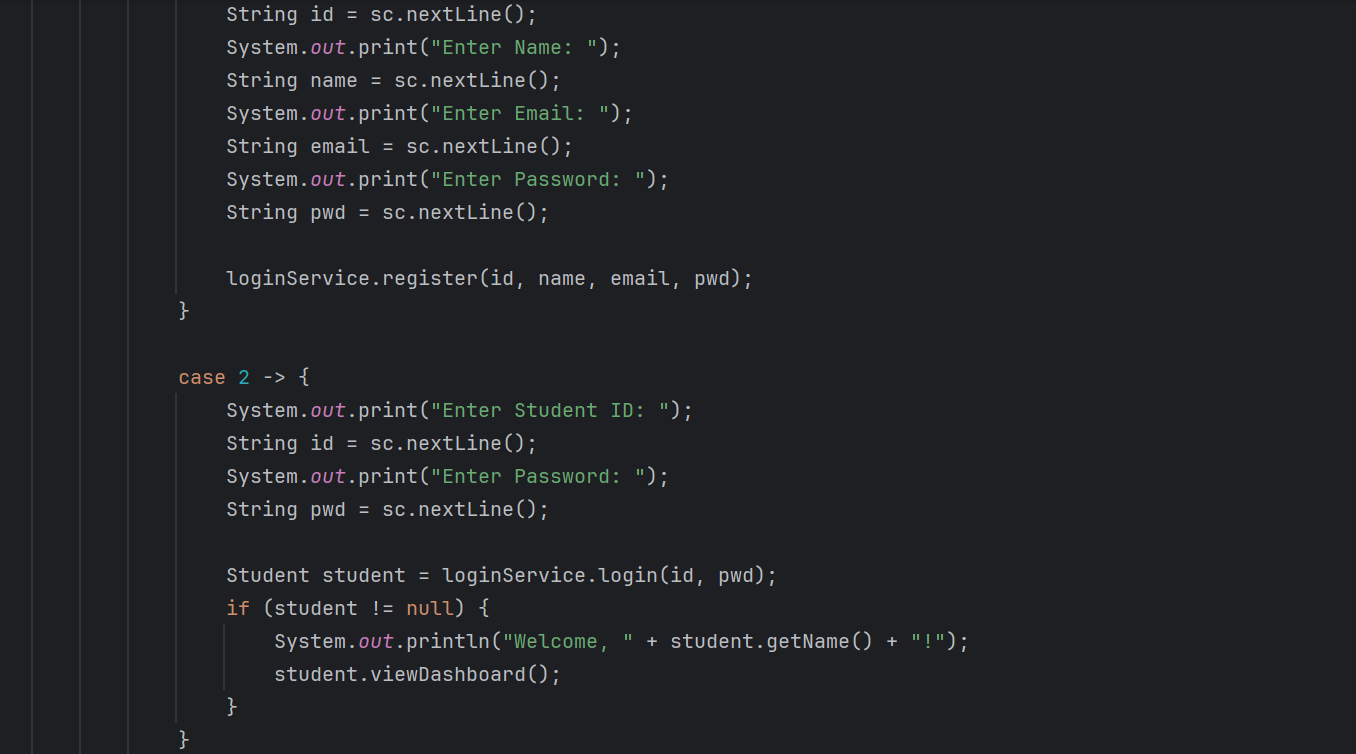


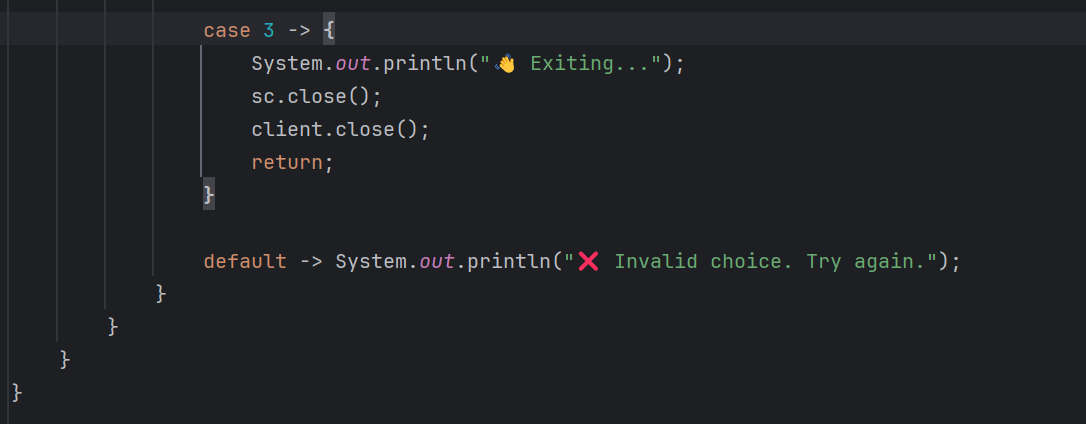


**StudentLoginMain.java**

import software.amazon.awssdk.services.dynamodb.DynamoDbClient;  
import java.util.Scanner;  
  
public class StudentLoginMain7 {  
 public static void main(String[] args) {  
 // Connect to DynamoDB Local using your config  
 DynamoDbClient client = DynamoDBConfig.*getClient*();  
 StudentDAO dao = new StudentDAO(client);  
 StudentLoginService loginService = new StudentLoginService(dao);  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.println("=== Student Login Portal (DynamoDB Local) ===");  
 while (true) {  
 System.*out*.println("\n1. Register");  
 System.*out*.println("2. Login");  
 System.*out*.println("3. Exit");  
 System.*out*.print("Enter choice: ");  
  
 int choice = sc.nextInt();  
 sc.nextLine(); // consume newline  
  
 switch (choice) {  
 case 1 -> {  
 System.*out*.print("Enter Student ID: ");  
 String id = sc.nextLine();  
 System.*out*.print("Enter Name: ");  
 String name = sc.nextLine();  
 System.*out*.print("Enter Email: ");  
 String email = sc.nextLine();  
 System.*out*.print("Enter Password: ");  
 String pwd = sc.nextLine();  
  
 loginService.register(id, name, email, pwd);  
 }  
  
 case 2 -> {  
 System.*out*.print("Enter Student ID: ");  
 String id = sc.nextLine();  
 System.*out*.print("Enter Password: ");  
 String pwd = sc.nextLine();  
  
 Student student = loginService.login(id, pwd);  
 if (student != null) {  
 System.*out*.println("Welcome, " + student.getName() + "!");  
 student.viewDashboard();  
 }  
 }  
  
 case 3 -> {  
 System.*out*.println("👋 Exiting...");  
 sc.close();  
 client.close();  
 return;  
 }  
  
 default -> System.*out*.println("❌ Invalid choice. Try again.");  
 }  
 }  
 }  
}







**StudentLoginService**

Handles the **business logic** for students:

* **register(studentId, name, email, password)**
  + Checks if the student ID already exists in DynamoDB.
  + If not, creates a new Student object and saves it.
  + Prints success or failure messages.
* **login(studentId, password)**
  + Checks if student exists in DynamoDB.
  + Validates password.
  + Returns the Student object if successful.
  + Prints appropriate messages for invalid ID or password.

**StudentLoginMain7**

Provides a **console-based UI** to interact with the service:

1. **Menu Options**
   * 1. Register → prompts for student details and registers them.
   * 2. Login → prompts for student ID & password and logs in.
   * 3. Exit → closes scanner & DynamoDB client.
2. **User Input**
   * Uses Scanner to read input.
   * Switch-case handles each option.
3. **Integration**
   * Creates a DynamoDbClient and StudentDAO.
   * Uses StudentLoginService to perform operations.
   * After login, the student can see their **dashboard** via student.viewDashboard().

**Key Features**

* Prevents duplicate student registrations.
* Provides clear console messages for success and errors.
* Cleanly integrates **persistent storage** with DynamoDB.
* Console interface allows **continuous operations until exit**.