KARPAGAM
COLLEGE OF ENGINEERING
Rediscover | Refine | Redefine
(Autonomous)

21PE04/21YE04 /21DE04/21SE04/21FE04

ADVANCED JAVA PROGRAMMING

ASSIGNMENT -1

SEMESTER- IV (2024)

BALAJI V(717822D107)

SARVESH S(717822D148)

DEPARTMENT OF

COMPUTER SCIENCE AND DESIGN

ABSTRACT

The University Enrollment System project aims to streamline the process of managing student enrollment, courses, and academic information in a university setting. This report provides an overview of the project, including its purpose and main functionalities. The system facilitates tasks such as adding, updating, and deleting student and course information, as well as managing enrollment and academic records. The report also covers the system's requirements analysis, design, implementation, and references to resources used.

Our system provides a comprehensive solution for managing student enrollments, courses, and faculty assignments efficiently. Administrators can effortlessly add or remove courses, enroll students, and assign faculty members via a user-friendly console interface. With robust CRUD operations implemented for each entity, including the capability to enroll and unenroll students from courses, educational institutions can maintain accurate records and streamline administrative tasks seamlessly.

With the advent of Information Technology in the last decade, the major focus has shifted from manual systems to computerised systems. Various systems viz. railway reservation, hospital management etc. involving manual work have been automated efficiently. Student course registration process in colleges involve filling registration forms manually, getting it signed by respective subject teachers, and then getting the documents acknowledged from the concerned Advisors, College Deans and Accounts Officers respectively. Finally the registration forms are submitted in the Administrative Branch. As is evident, this process is very laborious and time consuming. An Online Student Course Registration System has been developed to simplify the current manual procedure.This system has been developed using PHP, jQuery, Apache and MySQL. The front-end is designed using PHP with excerpts of code written using jQuery and back-end is designed and managed through MySQL. This system software is more secured, user-friendly and less time-consuming.

TABLE CONTENTS

INTRODUCTION

Background:
The University Enrollment System project is designed to address the challenges faced by universities in managing student enrollment and academic information efficiently. With the increasing complexity of educational systems, there is a growing need for digital solutions to streamline administrative tasks and enhance academic processes.

Objectives:

- Centralize student records, course information, and academic data in a database.
- Provide functionalities for CRUD operations on student and course records.
- Enable enrollment management for students and courses.
- Develop a user-friendly interface for students, faculty, and administrators.

2. System Analysis:

● Requirements analysis:

The requirements for the University Enrollment System were gathered through discussions with university administrators, faculty members, and students. Key features include:

- Database design for Students, Courses, Enrollments, and Academic Records.
- CRUD operations for managing student and course records.
- Enrollment management for student enrollment in courses.
- Academic record management for tracking student progress and grades.

● System specifications:

- Language: Java
- Database: MySQL
- JDBC for database connectivity
- Console interface for user interaction

3. System Design:

● Database schema design:

The database schema includes tables for Students, Courses, Enrollments, and Academic Records, with appropriate relationships and constraints.

● Application architecture:

The application follows a modular architecture with separate components for data access, business logic, and user interface.

4. Implementation:

- Environment Setup:

- MySQL database setup
- JDBC configuration for Java connectivity

JAVA CODE:

```
importjava.sql.*;
public class UniversityEnrollmentSystem {
private static final String URL = "jdbc:mysql://localhost:3306/university";
private static final String USERNAME = "your_username";
private static final String PASSWORD = "your_password";
private static Connection connection;
private static Statement statement;
private static ResultSetresultSet;
public static void main(String[] args) {
try {
connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
statement = connection.createStatement();
 scanner sc= new scanner (system.in);
createTables();
int choice;
do {
System.out.println("University Enrollment System");
System.out.println("1. Add Student");
System.out.println("2. Add Course");
System.out.println("3. View Students");
System.out.println("4. View Courses");
System.out.println("0. Exit");
System.out.print("Enter your choice: ");
choice = Integer.parseInt(sc.readLine());
switch (choice) {
case 1:
addStudent();
break;
case 2:
```

```
addCourse();
break;
case 3:
viewStudents();
break;
case 4:
viewCourses();
break;
case 0:
System.out.println("Exiting...");
break;
default:
System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 0);
    } catch (SQLException e) {
e.printStackTrace();
    } finally {
try {
if (resultSet != null) resultSet.close();
if (statement != null) statement.close();
if (connection != null) connection.close();
        } catch (SQLException e) {
e.printStackTrace();
        }
    }
  }
private static void createTables() throws SQLException {
    String createStudentsTable = "CREATE TABLE IF NOT EXISTS students (" +
        "id INT AUTO_INCREMENT PRIMARY KEY," +
        "name VARCHAR(255)," +
        "age INT)";
```

```
        String createCoursesTable = "CREATE TABLE IF NOT EXISTS courses (" +
                "id INT AUTO_INCREMENT PRIMARY KEY," +
                "name VARCHAR(255)," +
                "faculty VARCHAR(255))";
statement.executeUpdate(createStudentsTable);
statement.executeUpdate(createCoursesTable);
   }
private static void addStudent() throws SQLException {
System.out.print("Enter student name: ");
  Scanner sc= new Scanner (System.in);
        String name = sc.readLine();
System.out.print("Enter student age: ");
int age = Integer.parseInt(sc.readLine());
        String query = "INSERT INTO students (name, age) VALUES (?, ?)";
PreparedStatementpreparedStatement = connection.prepareStatement(query);
preparedStatement.setString(1, name);
preparedStatement.setInt(2, age);
preparedStatement.executeUpdate();
System.out.println("Student added successfully.");
   }
private static void addCourse() throws SQLException {
System.out.print("Enter course name: ");
  Scanner sc= new Scanner (System.in);
 String name = sc.readLine();
System.out.print("Enter faculty: ");
        String faculty = sc.readLine();
        String query = "INSERT INTO courses (name, faculty) VALUES (?, ?)";
PreparedStatementpreparedStatement = connection.prepareStatement(query);
preparedStatement.setString(1, name);
preparedStatement.setString(2, faculty);
preparedStatement.executeUpdate();
System.out.println("Course added successfully.");
```

```
    }
private static void viewStudents() throws SQLException {
        String query = "SELECT * FROM students";
resultSet = statement.executeQuery(query);
while (resultSet.next()) {
System.out.println("ID: " + resultSet.getInt("id") +
                ", Name: " + resultSet.getString("name") +
                ", Age: " + resultSet.getInt("age"));
        }
    }
private static void viewCourses() throws SQLException {
        String query = "SELECT * FROM courses";
resultSet = statement.executeQuery(query);
while (resultSet.next()) {
System.out.println("ID: " + resultSet.getInt("id") +
                ", Name: " + resultSet.getString("name") +
                ", Faculty: " + resultSet.getString("faculty"));
        }
    }
}
```

Mysql code:

```
CREATE DATABASE university;
USE university;
CREATE TABLE students (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255),
age INT);
CREATE TABLE courses (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255),
faculty VARCHAR(255)
);
```

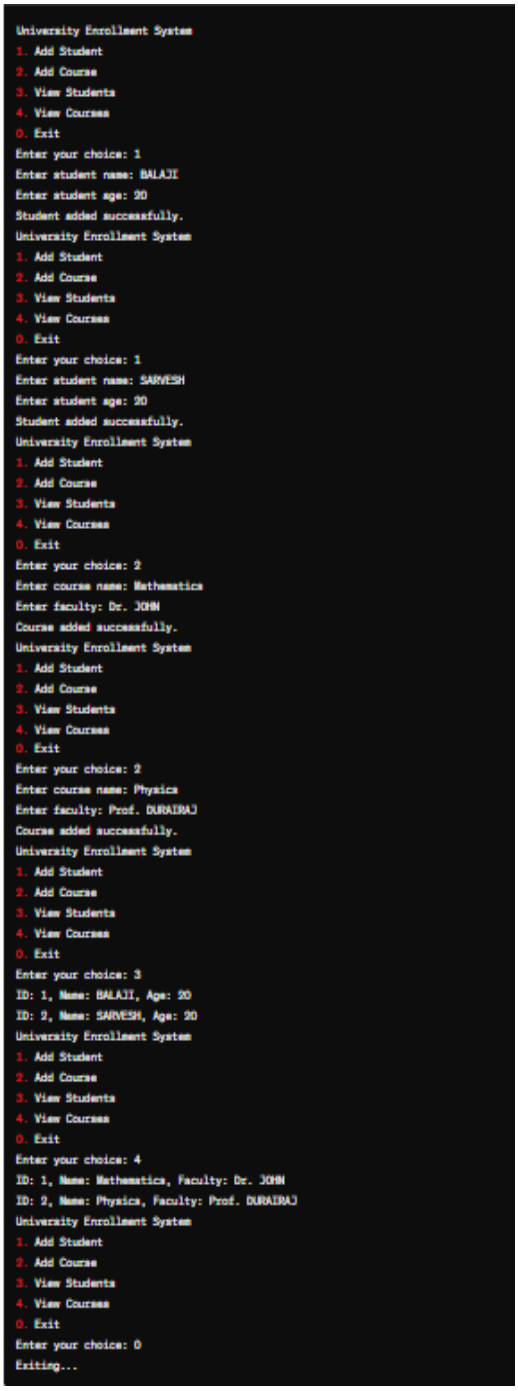
INSERT INTO students (name, age) VALUES ('Balaji,20);

INSERT INTO students (name, age) VALUES ('Sarvesh',20);

INSERT INTO courses (name, faculty) VALUES ('Mathematics', 'Dr. john');

INSERT INTO courses (name, faculty) VALUES ('Physics', 'Prof. Durairaj');

OUTPUT:



References:

- Books:

- "Database Systems: The Complete Book" by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom
- "Java: A Beginner's Guide" by Herbert Schildt

- Websites:

- Oracle's JDBC Documentation: https://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/
- MySQL Documentation: https://dev.mysql.com/doc/

- Tutorials and Guides:

- JavaTpoint JDBC Tutorial: https://www.javatpoint.com/example-to-connect-to-the-mysql-database
- MySQL Tutorial by TutorialsPoint: https://www.tutorialspoint.com/mysql/index.htm
- W3Schools SQL Tutorial: https://www.w3schools.com/sql/