C-DAC MUMBAI

Exam: MS .NET

Duration: 1 Hour 45 Mintues Marks: 15

Title: Online Course Enrollment System using ASP.NET Core MVC

Description: The goal of this project is to develop a web application for managing courses and student enrollments. The system will enable users to perform CRUD operations on courses and students and allow students to enroll in available courses. The Service and Repository patterns will ensure a scalable and maintainable architecture.

Functional Requirements:

- Course Model: CourseId, Name, Description, Duration (in weeks), Capacity, AvailableSeats
- Student Model: StudentId, Name, Email, ContactNumber
- Enrollment Model: EnrollmentId, StudentId (foreign key), CourseId (foreign key)

Repositories:

- ICourseRepository: Add, Update, Delete, GetCourseById, GetAllCourses.
- IStudentRepository: Add, Update, Delete, GetStudentById, GetAllStudents.
- IEnrollmentRepository: AddEnrollment, GetEnrollmentsByCourse, GetEnrollmentsByStudent.

Service Layer:

- EnrollmentService: Manages enrollment logic and capacity checks.

Controller Layer:

- CourseController: Handles course-related actions.
- StudentController: Handles student-related actions.
- EnrollmentController: Manages enrollment operations.

Desired Features:

- Course Management:
  - Add, view, edit, and delete courses.
  - Display course details such as name, description, duration, and remaining capacity.
- Student Management:
  - Add, view, edit, and delete student records.
- Enroll Students:
  - Allow students to enroll in courses.
  - Prevent over-enrollment by checking course capacity.
- View Enrollments:
  - View the list of students enrolled in a specific course.
  - View a student's enrolled courses.

Architecture:

MVC (Model-View-Controller):

- Model: Represents Courses, Students, and Enrollments.
- View: Provides UI for managing courses, students, and enrollments.

- Controller: Handles HTTP requests and invokes services.

Repository Pattern:

- ICourseRepository: CRUD operations for courses.
- IStudentRepository: CRUD operations for students.
- IEnrollmentRepository: Manages student-course relationships.

Service Layer:

- EnrollmentService: Handles business logic, such as checking course capacity and enrolling students.

Steps to Implement:

- Set up an ASP.NET Core MVC project.
- Define models for Course, Student, and Enrollment.
- Implement repositories for data access.
- Create service classes for business logic.
- Develop controllers and actions for managing courses, students, and enrollments.
- Design Razor Views for user interfaces.
- Add input validation and error handling.
- Test the application for all scenarios.

Deliverables:

- Fully functional Online Course Enrollment System.
- Clean, maintainable code using Repository and Service patterns.
- Comprehensive documentation and setup instructions.