

Final Milestone Assessment (60 Marks | 120 Minutes)

Format: User Story + Practical Tasks

Complexity Level: Average

Target Skills: React, Node.js, Express.js, MongoDB, REST API, Validation, Middleware, Authentication, Testing, Deployment Readiness

Problem Statement: SmartLearn Course Enrollment Portal

A mid-size EdTech platform, SmartLearn, is building a new online portal where learners can browse courses, enroll, track progress, and manage their accounts. You are hired as a MERN developer to implement critical user stories for this milestone.

SmartLearn has two major workflow segments:

1. **Segment A: Backend API (Node.js + Express + MongoDB)**
2. **Segment B: Frontend Interface (React)**

Each user story maps to mandatory tasks.

Total Marks = **60 Marks (Rubric attached below)**.

Segment A – Backend User Stories (30 Marks)

User Story A1 – Course Management API (15 Marks)

As an Admin, I want to create and fetch courses so that learners can browse the course catalog.

Tasks

1. **Design a Mongoose Course model** with the following fields:
 - courseId (string, unique, required)
 - title (string, required)
 - category (string, required)
 - price (number, required, min 0)
 - createdAt (date, default: current date)
2. **Implement the following API routes using Express:**
 - POST /api/courses
 - GET /api/courses
3. **Add validation using Joi or express-validator** for POST request.
4. **Implement an error-handling middleware** that handles validation errors, server errors, and invalid routes.

Respond with clean JSON structure:

```
{
  success: true/false,
  data: {},
  message: ""
}
```

User Story A2 – Basic Enrollment API (15 Marks)

As a learner, I want to enroll in a course so that I can start accessing it.

Tasks

1. Create an Enrollment model with fields:

- userId
- courseId
- enrolledOn (default: now)

2. Create route:

- POST /api/enroll

3. Validate:

- userId required
- courseId must exist in Course collection
- prevent duplicate enrollments

4. Write a small test using Mocha + Chai or SuperTest that verifies:

- API returns 201 on successful enrollment
 - Duplicate enrollment returns 400
-

Segment B – Frontend User Stories (30 Marks)

User Story B1 – Course Catalog UI (15 Marks)

As a learner, I want to view all available courses so I can choose one to enroll in.

Tasks

1. Create a React component CourseList.js
2. Fetch courses from /api/courses using useEffect
3. Display title, category, and price in a styled UI list
4. Add a button “Enroll Now” for each course
5. Show a loading and error state

User Story B2 – Enrollment Flow UI (15 Marks)

As a learner, I want to enroll in a course from the UI and receive feedback on the action.

Tasks

1. On clicking “Enroll Now,” send POST request to `/api/enroll`
2. Show appropriate toast/message on:
 - Success
 - Duplicate enrollment
 - Network error
3. Maintain a simple React state `enrolledCourses` to track enrolled items
4. Display enrolled course list below the catalog

Rubrics (60 Marks Total)

Backend (30 Marks)

Criteria	Marks
Database Schema Accuracy	5
API Implementation (CRUD + Validation)	10
Error Handling Middleware	5
Enrollment Logic + Duplicate Prevention	5
API Testing (Mocha/Chai/SuperTest)	5

Frontend (30 Marks)

Criteria	Marks
API Integration with React	10
Component Structure & State Management	5
UI Rendering of Course List	5
Enrollment Flow Working	5
Error/Loading States & Clean UX	5

Deliverables

Submit a ZIP containing:

Backend Folder

- Models (Course.js, Enrollment.js)
- Routes
- Middleware
- Test files
- Postman collection (optional)

Frontend Folder

- CourseList.js
- Enrollment components/pages

- API integration service (optional)

Readme File

Must include:

- Project setup instructions
- API endpoints
- Screenshots
- Instructions to run frontend and backend