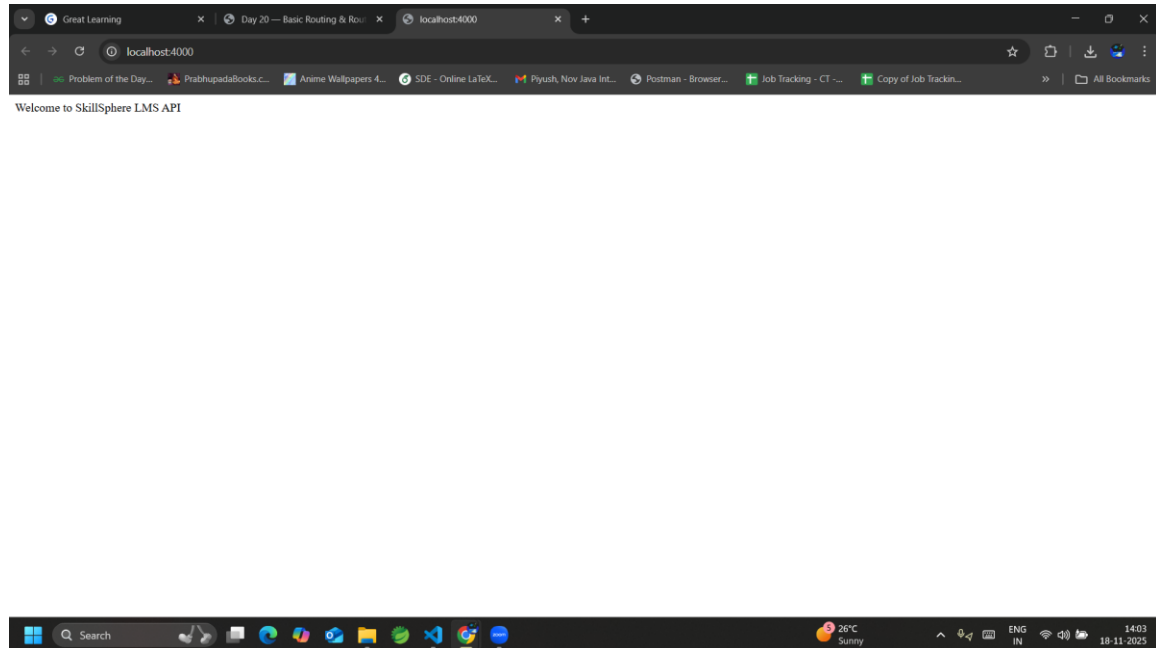


SkillSphere LMS – Day 20 Routing & Middleware Report

Challenge 1: Root Welcome Message

The root route '/' returns a welcome message as plain text. This confirms that the server is running and routing is configured properly.

Output:

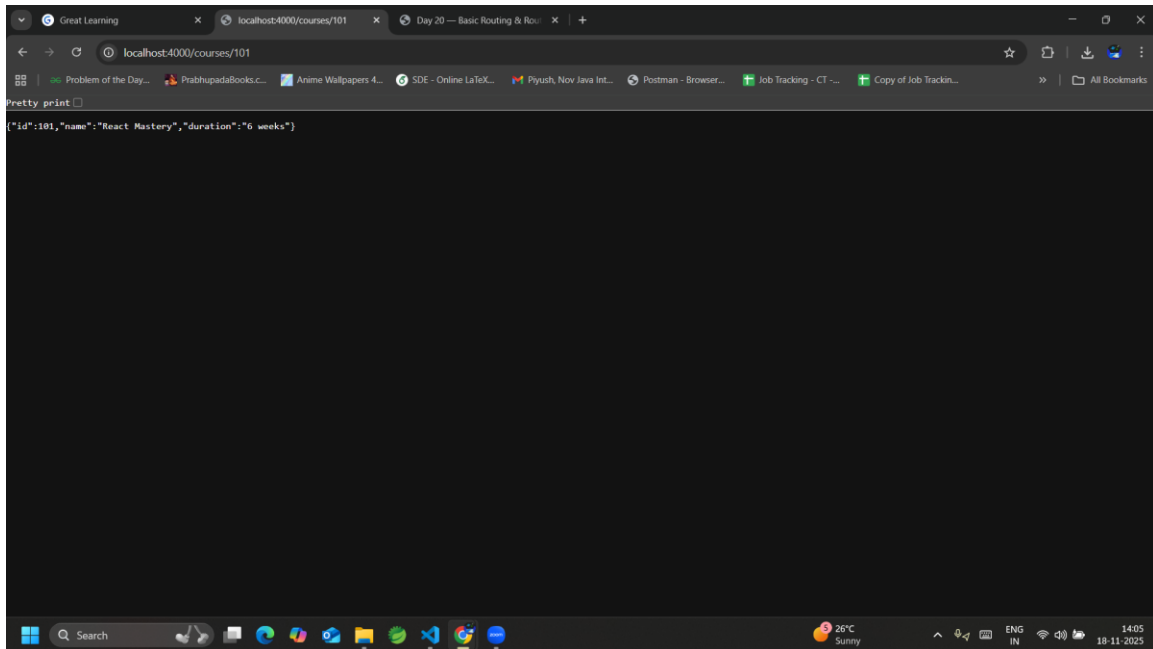


Challenge 2: Dynamic Course Route

Endpoint: GET /courses/:id

This route returns course details from courses.json based on the provided course ID.

Example Output (for ID = 101):



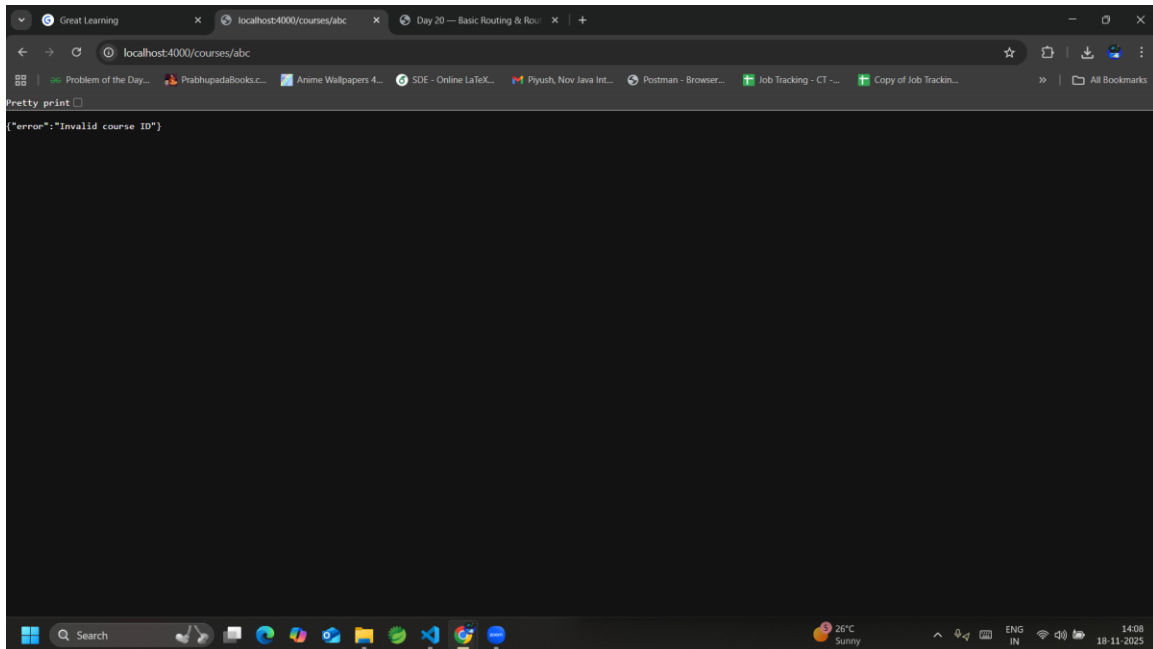
Key Code Snippet:

```
router.get('/:id', validateCourseId, (req, res) => {  
  const { id } = req.params;  
  
  // find the course with matching id (convert id to number)  
  const course = coursesData.courses.find(c => c.id === parseInt(id));  
  
  if (!course) {  
    return res.status(404).json({ error: "Course not found" });  
  }  
  
  res.json(course);  
});
```

Challenge 3: Route Middleware – Validate Course ID

A middleware function ensures that the course ID contains only numeric characters. If invalid, the server responds with an error message.

Output:



Middleware Code Snippet:

```
module.exports = function validateCourseId(req, res, next) {  
  
  const { id } = req.params;  
  // Check numeric ID: only digits allowed  
  if (!/^\d+$/.test(id)) {  
    return res.status(400).json({ error: 'Invalid course ID' });  
  }  
  // Proceed to the route handler  
  next();  
};
```

Approach Summary

- Implemented routing using Express.js following best practices.
- Separated route logic, middleware, and static JSON data for clean architecture.
- Used middleware to validate route parameters before reaching the handler.
- Ensured JSON responses for API endpoints as required.