

FullStack Lab 2 Report

-Kavish Solanki

This lab was much harder compared to lab 1 because we had to work with React framework which is a completely new concept to me. Working on this was fun as well as a headache because sorting out bugs and errors would take quite a while.

Challenges-

The biggest challenge I faced was with managing the put requests in the server and getting the react framework to work. I had to install webpack and babel for my project to work, I also tried vite but that didn't work well as there were always some bugs or errors. My biggest issue was with the PUT requests because the code was being run just fine with no errors but when I tried to pass a PUT command using thunderclient it just wouldn't go through. I scratched my head over this for a while and after hours of working and looking around over the internet I finally got it to work. But there are some random errors sometime where the PUT command won't go through, but if I restart the app then it works completely fine. Below are the screenshots for the code of PUT functions:

```
// Add a new project
router.post('/projects', async (req, res) => {
  try {
    const { project_code, project_name, project_description } = req.body;
    if (!project_code) {
      return res.status(400).json({ message: "Please enter project code" });
    }
    const createproj = await Project.create({
      project_code,
      project_name,
      project_description
    });
    res.status(201).json({
      message: "Project Added",
      Project: createproj.toObject({ getters: true, versionKey: false })
    });
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
});
```

```
// Add a new employee
router.post('/employees', async (req, res) => {
  try {
    const { employee_id, full_name, email, hashed_password } = req.body;
    if (!employee_id) {
      return res.status(400).json({ message: "Please enter Employee ID" });
    }
    const createemp = await Employee.create({
      employee_id,
      full_name,
      email,
      hashed_password
    });
    res.status(201).json({
      message: "Employee created successfully",
      employee: createemp.toObject({ getters: true, versionKey: false })
    });
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
});
```

```
// Add a new project assignment
router.post('/project_assignments', async (req, res) => {
  try {
    const { employee_id, project_code, start_date } = req.body;

    const existingEmployee = await Employee.findOne({ employee_id });
    if (!existingEmployee) {
      return res.status(400).json({ message: "Employee not found" });
    }

    const existingProject = await Project.findOne({ project_code });
    if (!existingProject) {
      return res.status(400).json({ message: "Project not found" });
    }

    const newAssignment = new ProjectAssignment({
      employee_id: existingEmployee.employee_id,
      project_code: existingProject.project_code,
      start_date
    });

    await newAssignment.save();

    res.status(201).json({ message: "Project assignment saved successfully", newAssignment });
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
});
```

My biggest issue was with /project_assignments where it will sometimes take the input but when I change the values to something else it keeps failing, and then I restart the code and suddenly it works well.

Below is the link for the project:

https://github.com/Kavish1202/Fullstack_Lab2