LONDON METROPOLITAN UNIVERSITY

PROFESSIONAL WORK PLACEMENT

LEARNING LOG

ID: 22072081

NAME: Anjal Rai

COURSE: BSc (Hons) Computing

CDL TUTOR: Nishesh Bishwas

PLACE OF WORK (Company Name): Hunchha Digital

DATES OF ENTRIES IN LEARNING LOG:

Log Index	Start Date	End Date	Company Supervisor Signature
Log 1	25 June, 2024	4 July, 2024	
Log 2	5 July, 2024	15 July, 2024	

Log 3	16 July, 2024	28 July, 2024	
Log 4	29 July, 2024	9 August, 2024	
Log 5	11 August, 2024	19 August, 2024	
Log 6	20 August, 2024	30 August, 2024	
LO7	31 August, 2024	12 September, 2024	
LO8	13 September, 2024	24 September, 2024	

LEARNING LOG 1

For the period 24/06/2024 to 04/07/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Efficient Workload Management and API Development

During the first week, I was introduced to my supervisor and assigned tasks to work on independently. I effectively managed my workload by developing server-side logic and creating RESTful APIs. I utilized the Thunder Client extension in VSCode for testing and prioritized tasks to meet deadlines.

LO3: Adapting to New Tools and Responsibilities

I adapted by mastering tools like Sequelize ORM and the Thunder Client extension in VSCode, which I used for testing and managing API development. I efficiently handled my workload by developing RESTful APIs and troubleshooting backend issues, incorporating feedback from my supervisor to enhance the project. My communication skills improved through regular progress updates and presentations. I successfully implemented MVC architecture and user authentication with Express.js, This adaptability allowed me to meet project demands and contribute effectively to the backend functionality.

LO4: Troubleshooting and Enhancing Backend Application Quality

I focused on troubleshooting issues within Node.js applications, using debugging tools to resolve errors in backend logic and API functionality. Feedback from my supervisor was crucial for finding solutions and enhancing the application’s quality.

LO5: Effective Communication and Project Status Presentations

I enhanced my communication skills by providing regular updates on my development progress to my supervisor and discussing any challenges encountered. I delivered a presentation on the status of backend development, detailing the features implemented and the current project status. Incorporating feedback from my supervisor helped refine the backend functionalities and improve the project’s overall quality.

LO6: Mastering RESTful API Development and MVC Authentication

I participated in a comprehensive Node.js training program that laid the foundation for creating and managing RESTful APIs. I applied these skills by developing a scalable backend system, testing API functionality using the Thunder Client extension in VSCode, and implementing the MVC (Model-View-Controller) architecture to structure the application efficiently.

LO7: Implementing API Routes and Authentication

I utilized Express.js to develop and manage various API routes in the Node.js application. This included implementing user authentication and authorization with JSON Web Tokens (JWT), as well as creating endpoints for user login, registration, profile management, and data updates. Although I did not create custom middleware, I effectively integrated Express.js with other components, ensuring efficient handling of HTTP requests and responses.

LO8: Optimizing Database Operations and Entity Relationships with Sequelize

I employed Sequelize ORM to manage database operations, including performing CRUD operations and defining database schemas. I also set up and maintained relationships between data entities and ensured data integrity. My work with Sequelize facilitated efficient database interactions and contributed significantly to the overall functionality of the backend.

What I did well (Refer to skills used)

Effective Time Management and Independent Task Execution

During the first week, I effectively managed my time by independently working on tasks assigned by my supervisor. I developed server-side logic and created RESTful APIs, ensuring that I met deadlines and kept my workload organized.

Resolving Backend Issues and Enhancing Application Quality

I demonstrated strong problem-solving skills by focusing on troubleshooting issues within Node.js applications. Using various debugging tools, I resolved errors in backend logic and API functionality. The constructive feedback from my supervisor played a crucial role in refining my solutions and improving the overall quality of the application.

Applying Node.js Knowledge to Develop Scalable Backend Systems

I successfully applied the foundational knowledge gained from a comprehensive Node.js training program. I developed a scalable backend system, managed RESTful APIs, and used the Thunder Client extension for testing. Implementing the MVC (Model-ViewController) architecture helped in structuring the application efficiently, showcasing my ability to apply theoretical knowledge to practical tasks.

What I could improve on (i.e. skills I want to improve)

Enhance Communication by Seeking Frequent Feedback

Although I have been providing updates and participating in discussions, I could further improve by actively seeking feedback more frequently and clarifying any uncertainties during discussions to ensure clearer understanding and communication.

Gaining Expertise in Creating Custom Middleware

I have yet to create custom middleware, which is a crucial aspect of using Express.js. Gaining experience in this area will enhance my ability to manage and control request handling and improve the functionality and security of my applications.

Expanding Knowledge of Advanced Sequelize Features for Complex Database Management

While I have used Sequelize for basic CRUD operations, I need to extend my understanding of its advanced features. Expanding my knowledge in this area will enable me to handle more complex database interaction

Action I can take to improve my skills and learning (make these “SMART”):

Enhance Communication by Seeking Frequent Feedback

Goal: Improve feedback integration and clarity in communication.

Action: Schedule bi-weekly feedback sessions with my supervisor and prepare questions in advance.

Timing: Implement starting next week and continue throughout the internship.

Gaining Expertise in Creating Custom Middleware

Goal: Gain experience in creating and integrating custom middleware.

Action: Develop one custom middleware function for a specific use case by the end of the next two weeks.

Timing: Complete development and integration by the end of the second week.

Expanding Knowledge of Advanced Sequelize Features for Complex Database Management

Goal: Enhance my skills in advanced Sequelize features.

Action: Study advanced Sequelize documentation and tutorials to understand complex features.

Timing: Allocate two hours each week for this study.

LEARNING LOG 2

For the period 05/07/2024 to 15/07/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Efficient Time Management and Task Prioritization

This week, I managed my time effectively by prioritizing tasks for backend development, authorization features, and API integration for an e-commerce website and a to-do list application. I followed a structured approach and communicated regularly with my supervisor to ensure timely task completion.

LO4: Overcoming Technical Challenges and Enhancing Middleware

I faced technical challenges with Sequelize ORM and the Multer library. I created two middleware functions—one for token and role verification and another for file uploads using Multer. I also learned to establish relationships between database tables using Sequelize. I resolved issues by consulting my supervisor, reading documentation, and using Stack Overflow.

LO5: Strengthening Communication through Updates and Presentations

I continued to enhance my communication skills by regularly updating my supervisor on project progress and discussing technical challenges. I delivered presentations on the backend development of the e-commerce website. Feedback from my supervisor helped refine the backend functionalities and improve the overall quality of the projects.

LO6: Building and Managing Backend Systems with Node.js

I applied my Node.js skills to develop backend systems for an e-commerce website and a to-do list project. This involved creating and managing RESTful APIs, implementing middleware for JWT token verification and role checking, and utilizing the MVCR architecture for efficient application structuring. I also used the Multer library to handle file uploads and management.

LO7: Implementing Secure API Routes with Express.js

I utilized Express.js to handle various API routes and user authentication processes. I implemented JWT tokens for secure user login and registration and created middleware to check tokens and user roles. This middleware ensured that only authorized users could access specific resources. My work with Express.js was crucial for managing HTTP requests, responses, and overall API functionality.

LO8: Managing Complex Databases with Sequelize ORM

I extensively used Sequelize ORM for CRUD operations, schema definition, and establishing relationships between data entities. This experience improved my ability to handle complex database interactions and maintain data integrity.

LO6: Mastering RESTful API Development and MVC Authentication

I participated in a comprehensive Node.js training program that laid the foundation for creating and managing RESTful APIs. I applied these skills by developing a scalable backend system, testing API functionality using the Thunder Client extension in VSCode, and implementing the MVC (Model-View-Controller) architecture to structure the application efficiently.

LO7: Implementing API Routes and Authentication

I utilized Express.js to develop and manage various API routes in the Node.js application. This included implementing user authentication and authorization with JSON Web Tokens (JWT), as well as creating endpoints for user login, registration, profile management, and data updates. Although I did not create custom middleware, I effectively integrated Express.js with other components, ensuring efficient handling of HTTP requests and responses.

LO8: Optimizing Database Operations and Entity Relationships with Sequelize

I employed Sequelize ORM to manage database operations, including performing CRUD operations and defining database schemas. I also set up and maintained relationships between data entities and ensured data integrity. My work with Sequelize facilitated efficient database interactions and contributed significantly to the overall functionality of the backend.

LO9: Version Control with Git

I used Git for version control to manage code changes and maintain a structured development process. I consistently committed and pushed changes, managed branches, and resolved merger conflicts.

What I did well (Refer to skills used)

Time Management and Task Prioritization

This week, I managed my time effectively by focusing on key tasks for the e-commerce website and the to-do list project. I prioritized backend development, authorization features, and API integration. My organized approach and consistent communication with my supervisor ensured that I stayed on track and met deadlines.

Problem-solving and Technical Skills

I encountered several technical challenges with Sequelize ORM and the Multer library, as well as in creating middleware for token verification and file uploads. I resolved these issues with guidance from my supervisor, documentation, and Stack Overflow, which enhanced my technical expertise. Additionally, I learned to set up table relationships with Sequelize and excelled in API creation and server setup.

Communication and Presentations

I maintained regular communication with my supervisor, providing updates and discussing challenges. I also delivered a presentation on the backend functionality of the e-commerce website and the to-do list project. The feedback I received helped refine my work and improve the overall project outcomes.

What I could improve on (i.e. skills I want to improve)

Completing the E-Commerce Backend Project

I have not completed the backend development for the e-commerce project. To enhance my skills and project delivery, I plan to focus on finalizing this project. Completing it will help me apply and solidify my backend development knowledge.

Collaboration with Front-End Developers

Currently, I have not worked collaboratively with front-end developers. To improve in this area, I plan to start working with a front-end developer on future projects. This experience will help me understand the integration of front-end and back-end components and improve my teamwork skills.

Improving Problem-Solving Skills

I aim to enhance my problem-solving skills by tackling more complex issues and applying advanced debugging techniques. This includes practicing troubleshooting strategies and learning from case studies.

Applying Other Node.js Frameworks

I want to explore and apply other Node.js frameworks to broaden my technical skill set. This will involve studying different frameworks and implementing them in projects to gain practical experience.

Action I can take to improve my skills and learning (make these “SMART”):

Completing the E-commerce Backend Project

Goal: Finalize the backend for the e-commerce project by addressing remaining features and issues.

Action: Allocate dedicated time to complete all planned features, conduct thorough testing, and address any bugs or incomplete tasks.

Timing: Aim to complete the project within this week, with a deadline of 7 July 2024.

Collaborating with Front-End Developers

Goal: Start collaborating with front-end developers to integrate front-end components with backend systems.

Action: Schedule and participate in regular meetings or collaborative sessions with frontend developers. Work on integration tasks together.

Timing: Initiate collaboration within the next two weeks and continue throughout future projects.

Improving Problem-Solving Skills

Goal: Enhance problem-solving skills by practicing advanced debugging techniques and learning from case studies.

Action: Practice troubleshooting with challenging scenarios and study case studies on advanced problem-solving techniques.

Timing: Engage in these activities weekly, setting aside time to work on problem-solving exercises and review case studies.

Applying Other Node.js Frameworks

Goal: Explore and apply other Node.js frameworks to broaden technical skills.

Action: Study tutorials and documentation for a new Node.js framework and implement it in a project.

Timing: Begin exploration within the next month and apply new frameworks in a project by the end of the month.

LEARNING LOG 3

For the period 16/07/2024 to 28/07/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management in Multi-Project Development

During this period, I completed the backend development of the e-commerce website, utilizing the MVCR architecture as guided by my supervisor. My focus was on code reusability, studying various documentation on database schemas, and modules related to e-commerce websites, as well as handling requests with Sequelize. Additionally, I began working on a new project called the Blog Management System using EJS templating. Alongside these projects, I managed extra tasks such as creating APIs, optimizing database queries, and setting up user authentication. Despite the increased workload, I effectively prioritized tasks and met all deadlines, enhancing my time management skills.

LO2: Collaborative Integration and Team Coordination

I continued to collaborate closely with my supervisor and team members, especially during the development and deployment phases of my projects. I actively participated in team discussions, sought feedback, and incorporated suggestions into my work. Working closely with a front-end developer ensured seamless integration of backend functionality, including APIs and authentication. This collaboration eliminated the need for hosting the APIs and database separately, streamlining our workflow and aligning our efforts within the team.

LO3: Adapting to New Technologies and Project Requirements

I improved my adaptability by quickly learning and applying new technologies across various projects. I effectively used Sequelize for complex database management, including pagination and CRUD operations. I integrated Nodemailer for email functionalities in Node.js and resolved hosting issues. Adapting to the MVCR architecture as guided by my supervisor improved code reusability. I also embraced EJS for serverside rendering and frontend integration, meeting project requirements efficiently.

LO4: Resolving Technical Challenges and Enhancing Problem-Solving Skills

During my projects, I encountered technical challenges with Sequelize ORM, EJS templating, and hosting Node.js on Render and MySQL on Clever Cloud. I had difficulties making the frontend dynamic, applying CRUD operations, using external CSS and images, and implementing pagination. To overcome these issues, I sought solutions through online courses, YouTube tutorials, and guidance from my supervisor. Additionally, I conducted research, read documentation, and explored GitHub repositories to improve my problem-solving skills and strengthen my JavaScript fundamentals.

LO5: Maintaining Clear Communication and Detailed Documentation

I maintained regular communication with my supervisor, providing updates on project progress and discussing technical challenges. I delivered presentations on the backend development of the e-commerce website and the Blog Management System, effectively explaining complex technical details. To ensure smooth collaboration with the front-end developer, I frequently discussed API functionalities and the attributes of front-end forms. Using Postman, I created detailed API documentation, ensuring clear understanding for the front-end team.

LO6: Advancing Node.js Backend Development with RESTful APIs and EJS

I enhanced my Node.js expertise by developing the backend for an e-commerce website and a Blog Management System. This involved creating and managing RESTful APIs, integrating server-side rendering using EJS, and implementing the Nodemailer module for "forgot password" functionality. Additionally, I worked on optimizing server performance and implementing error handling to improve the overall stability of the applications.

LO7: Implementing Secure and Efficient Solutions with Express.js

In my work with Express.js, I handled critical tasks such as implementing pagination, token-based authentication, and managing server-side requests and responses. I focused on creating efficient and secure applications by handling URL redirection, configuring the view engine with EJS, and ensuring proper token management.

LO8: Database Management with Sequelize

I gained extensive experience with Sequelize, particularly in defining and managing relationships between tables, applying pagination, and performing CRUD operations. These skills allowed me to build robust data models and optimize database interactions effectively.

LO9: Managing Code with Git and GitHub for Collaborative Development

From the start, I created repositories and pushed my code to GitHub. Over time, I learned to collaborate effectively with other team members by using Git for version control. This collaboration included working with front-end developers, enhancing my skills in managing code changes, handling pull requests, and coordinating with team members.

What I did well (refer to skills used)

Technical Proficiency

I excelled in applying my technical skills across various projects, including managing complex database relationships, handling CRUD operations, and implementing pagination. My work with Express.js and Nodemailer demonstrated my ability to handle essential backend functionalities and resolve hosting issues.

Problem-Solving Abilities

I demonstrated strong problem-solving skills by addressing and overcoming technical challenges. I learned to utilize advanced features in Sequelize and Express.js and resolved issues with Nodemailer and hosting platforms. My approach to solving these problems ensured that project components functioned efficiently and met the necessary requirements.

Collaboration and Communication

I worked effectively with team members using Git and GitHub for version control. I coordinated with front-end developers to align our tasks and meet project goals. My clear communication and teamwork helped ensure that our integration efforts were successful and that project deadlines were met.

Adaptability

I quickly adapted to new tools and technologies, such as EJS for server-side rendering, Nodemailer for email functionalities, and pagination techniques. My ability to adjust to new requirements and tools allowed me to meet project needs effectively.

What I could improve on (i.e. skills I want to improve)

Deepening Node.js Expertise

My primary focus is to deepen my knowledge of Node.js by learning advanced concepts such as implementing real-time chat systems using Socket.IO. I plan to work on real-life projects that incorporate these technologies, which will help me apply new skills and meet project deadlines effectively.

Enhancing Deployment and Hosting Skills

I plan to strengthen my skills in deployment and hosting by mastering CI/CD pipelines and cloud infrastructure management. This includes gaining proficiency in deploying Node.js applications effectively and managing hosting environments to ensure reliability and scalability.

Improving Collaboration and Communication

I seek to enhance my collaboration skills by refining my strategies for working with team members on backend projects. This involves using version control systems like Git and GitHub more effectively and communicating clearly to ensure seamless integration and project success,

Action I can take to improve my skills and learning (make these "SMART"):

Deepening Node.js Expertise

Goal: Enhance Node.js skills by learning and implementing real-time chat systems using Socket. IO.

Action: Enroll in a course on Socket.IO, integrate it into a project, and complete a fully functional chat system by applying new knowledge.

Timing: Complete by 25 August 2024.

Enhancing Deployment and Hosting Skills

Goal: Improve deployment and hosting skills by mastering CI/CD pipelines and cloud infrastructure management.

Action: Set up and configure a CI/CD pipeline for a Node.js application, learn best practices for cloud deployment, and manage deployment environments effectively.

Timing: Complete the CI/CD setup and cloud management training within 3 weeks, with all configurations and management tasks finalized by 13 August 2024.

Improving Collaboration and Communication

Goal: Enhance collaboration skills by refining strategies for effective teamwork and version control.

Action: Schedule regular meetings with the supervisor to receive feedback and guidance. Regularly use Git for version control in ongoing projects. Additionally, enhance documentation practices by preparing detailed API documentation using Postman and delivering presentations on the work.

Timing: Implement improved practices by 20 August 2024.

LEARNING LOG 4

For the period 29/07/2024 to 09/08/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management

I continued working on the Blog Management Project while balancing multiple tasks, including refining database schemas, implementing UI features, and integrating payment systems. Effective time management ensured the timely completion of all deliverables, including the successful deployment of the project.

LO2: Teamwork and Collaboration

I collaborated closely with my supervisor and teammates, actively seeking feedback on design elements and schema refinement. Their insights guided improvements in table relationships and UI enhancements, ensuring alignment with user requirements and best practices.

LO3: Adaptability

I enhanced my adaptability by learning advanced EJS templating for server-side rendering and implementing pagination. Additionally, I tackled the integration of the Khalti payment gateway into the e-commerce project, overcoming challenges related to merchant account creation by utilizing my supervisor’s account.

LO4: Problem-Solving Skills

I resolved technical issues such as database schema optimization, UI design, and integrating real-time communication using WebSockets. By leveraging a paid course from Digital Pathsala, I improved my understanding of WebSockets, enabling effective real-time communication for future projects.

LO5: Communication Skills

I provided regular updates to my supervisor, including a presentation of the completed Blog Management Project. This presentation showcased the project's features, architecture, and real-time communication functionality, strengthening my ability to explain complex technical concepts clearly.

LO6: Node.js Backend Development

I deepened my expertise in Node.js by developing robust backend functionality for the Blog Management Project. This included creating and managing RESTful APIs, refining database interactions, and ensuring seamless integration with front-end components.

LO7: Express.js Development

Using Express.js, I implemented middleware, optimized routing, and handled secure payment integration for the e-commerce project. These tasks ensured that APIs were efficient, secure, and scalable.

LO8: Database Management with Sequelize

I explored advanced database relationships, refining schemas for the Blog Management Project. This included ensuring proper data integrity and designing efficient queries to support pagination and dynamic content rendering.

LO9: Version Control with Git

I consistently used Git for version control, managing code updates and ensuring clean collaboration with teammates. My GitHub repository reflected all project progress, and pull requests were handled efficiently

What I did well (refer to skills used)

Technical Proficiency:

Successfully applied advanced EJS templating, database relationships, and pagination techniques to improve the Blog Management Project.

Problem-Solving:

Overcame challenges with the Khalti payment integration and real-time communication by learning WebSockets through a structured course.

Team Collaboration:

Actively incorporated feedback from my supervisor and peers, refining the project to meet high-quality standards.

What I could improve on (i.e. skills I want to improve)

Advanced Payment Systems:

Learn to independently create merchant accounts and integrate other payment gateways.

WebSockets Proficiency:

Dive deeper into WebSocket debugging and optimization for real-time communication.

UI/UX Design:

Gain more knowledge about user-centered design principles to create more intuitive interfaces.

Action I can take to improve my skills and learning (make these "SMART"):

Mastering Payment Integrations:

Enroll in an online course on payment gateway integration.

Enhancing Real-Time Communication:

- Build a small-scale chat application using WebSockets.
- Complete by 25 August 2024.
- Improving UI/UX Design.

Study design principles and implement them in upcoming projects.

Review designs with feedback from friends and supervisors by 15 August 2024

LEARNING LOG 5

For the period 11/08/2024 to 19/08/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management

I focused on optimizing and restructuring my Node.js code to improve reusability and maintain a professional standard. Alongside this, I balanced multiple tasks, including learning WebSockets, completing the Blog Management System, and beginning backend development for the School Management System

LO2: Teamwork and Collaboration

I worked closely with my supervisor and frontend developers, especially while implementing authentication for the School Management System. Their guidance helped me refine my approach structuring APIs and improving backend efficiency.

LO3: Adaptability

I explored new concepts by diving into WebSockets through YouTube tutorials and documentation. After gaining and understanding, I applied it by integrating WebSockets into the Blog Management System. Additionally, I started learning React.js for frontend development with guidance from my supervisor and the frontend team.

LO4: Problem- Solving Skills

I faced difficulties implementing WebSockets correctly and understanding its real-time communication behavior. I also had challenges making EJS templates dynamic, especially when dealing with DOM manipulation. By experimenting with practice projects, debugging systematically, and referring to online documentation, I overcame these challenges.

LO5: Communication Skills

I regularly updated my supervisor on my work, including code optimization, WebSocket integration, and frontend-backend collaboration. Their feedback helped me improve API structuring and backend efficiency.

LO6: Node.js Backend Development

I successfully completed both the frontend and backend for the Blog Management System using Node.js and EJS. I then moved on to developing the backend for the School Management System, focusing on authentication.

LO7: Express.js Development

I implemented authentication mechanisms in the School Management System using Express.js, ensuring secure handling of user sessions while working with the frontend team.

LO8: Database Management with Sequelize

I designed and managed database models for the School Management System authentication module, refining schema structures to ensure data integrity.

LO9: Version Control with Git

I improved my use of Git by committing regularly and managing different branches for various features. This helped maintain a clean repository and smooth collaboration with the team.

What I did well (refer to skills used)

Technical Proficiency:

Successfully optimized and restructured backend code to improve reusability and maintainability.

Problem-Solving:

Successfully integrated WebSockets and overcame challenges in dynamic rendering with EJS.

Project Completion:

Completed both frontend and backend development for the Blog Management System.

Team Collaboration:

Effectively worked with the frontend developers and my supervisors to integrate authentication and API structuring.

What I could improve on (i.e. skills I want to improve)

WebSockets Debugging:

Gain deeper knowledge of WebSockets for better real-time functionality.

EJS Dynamic Rendering:

Improve DOM manipulation when using EJS templates.

Backend Code Professionalism:

Enchance API structuring and optimize query performance.

Consistent Version Control:

Ensure consistent and meaningful commit messages.

Action I can take to improve my skills and learning (make these "SMART"):

Enhancing Backend Code Quality

Review the best practices and get feedback from my supervisor.

Completing School Management System Backend

Develop 50-60% of the backend with structured APIs.

Maintaining GIT Version Control:

Regularly commit and use proper GIT practices.

LEARNING LOG 6

For the period 20/08/2024 to 30/08/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management

I allocated time to implement and separate backend layers using the MVC architecture. I focused on setting up routes, controllers, and models directories for modularity and scalability.

LO2: Teamwork and Collaboration

I communicated regularly with the team to confirm route requirements and role-based data access, helping align my Express routes and middleware with user expectations.

LO3: Adaptability

I learned how to apply the MVC structure effectively. I refactored controller logic to remove direct database queries from route files, improving separation of concerns.

LO4: Problem- Solving Skills

Faced middleware chaining issues when protecting routes using JWT. I resolved it by restructuring the authMiddleware.js file and testing role-based access in Postman.

LO5: Communication Skills

I presented my backend folder structure to the team and walked through how the routes pass control to middleware and controllers.

LO6: Node.js Backend Development

- Created modular route files such as userRoutes.js, studentRoutes.js, and teacherRoutes.js.
- Controllers handled logic such as registerUser, getProfile, and student record management.

- Used express.Router() to route requests cleanly.

LO7: Express.js Development

Built and registered middleware functions like authMiddleware.js to verify JWT tokens. Protected sensitive routes like /profile, /students, and /fees.

LO8: Database Management with Sequelize

Modeled User, Profile, Student, Teacher, and Parent tables. Implemented associations (e.g., Student.belongsTo(Parent), Teacher.belongsToMany(Student))

LO9: Version Control with Git

Maintained routes-refactor and auth-system branches. Used clear commit messages like “added token middleware” and “refactored student controller”.

What I did well (refer to skills used)

Modular MVC Integration

Successfully applied the MVC pattern to organize backend components. Split responsibilities across route files, controllers, and models, which led to better maintainability and reduced coupling.

Secure Route Middleware Setup

Implemented JWT-based middleware (authMiddleware.js) to protect routes. Resolved chaining and role-access issues through debugging and restructuring.

Clear Communication with Team

Effectively explained the structure and flow of requests across the backend layers during team discussions. Helped frontend teammates understand how to integrate securely with backend APIs.

Model Design and Relationship Mapping

Created relational models using Sequelize (e.g., User, Student, Teacher) and mapped associations to enable smooth querying and data fetching across the system

What I could improve on (i.e. skills I want to improve)

Authentication Logging:
Develop a logging system to record failed or unauthorized access attempts for auditing and debugging purposes.

Error Centralization:
Create a more unified error-handling middleware rather than scattering try/catch logic across all controllers.

API Structure Consistency:
Improve standardization of API response shapes for better frontend integration and developer experience.

Commit Strategy:
Improve granularity of commits and write more descriptive messages as project modules grow more complex.

Action I can take to improve my skills and learning (make these "SMART"):

Implement Failed Auth Logging (By 12 Sept 2024)
Integrate logging middleware that captures failed JWT verifications and stores them in a log file or DB.

Centralize Error Handling (By 12 Sept 2024)
Set up a global error handler to manage and standardize all error messages from controllers.

Standardize API Responses (By 15 Sept 2024)

Create a response format utility and apply it across all controller responses.

Improve Git Practices

Review and follow semantic commit message guidelines consistently across branches

LEARNING LOG 7

For the period 31/08/2024 to 12/09/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management

I planned the implementation of fee and expense modules and mapped the Sequelize models before developing their respective controllers and routes.

LO2: Teamwork and Collaboration

Worked with frontend developer to document and verify POST /fees and GET /students/:id/fees endpoints. Ensured responses followed consistent JSON structures.

LO3: Adaptability

Adapted to Sequelize's transaction system to handle batch fee assignments. Refactored routes to use controller methods for better clarity and testing.

LO4: Problem-Solving Skills

Resolved a bug where fees assigned via one controller endpoint weren't reflected due to missing associations. Used Sequelize 'include' option to verify relationships.

LO6: Node.js Backend Development

- Built controller methods for assignFee, getStudentFees, and createFeeType.
- Ensured studentController.js and feesController.js used async/await with proper error handling.

LO7: Express.js Development

Used authMiddleware to protect all financial routes. Ensured only admin users could POST or PUT data via route-level role checks.

LO8: Database Management with Sequelize

- Modeled Fee, FeeType, and Expense.
- Added foreign keys: fees.studentId, fees.feeTypeId.
- Used transactions to avoid partial saves when assigning multiple fees.

What I did well (refer to skills used)

Transactional Fee Handling

Utilized Sequelize transactions to assign multiple fees without breaking integrity. Troubleshot missing associations using Sequelize’s include, ensuring accurate database relations.

Scalable Financial Module Implementation

Built clean, role-protected routes and controllers for fee management. Maintained RESTful patterns to ensure consistent backend API design.

Collaboration and Testing Preparation

Worked closely with the Frontend team to verify endpoints like GET /students/:id/fees.

What I could improve on (i.e. skills I want to improve)

Model-Level Validations

While the models are functional, I can improve them by adding more robust validations, particularly for required fields and data types to prevent database inconsistencies.

Refining Controller Logic

There is some repetition in the controller logic for handling fees and expenses. I could refactor this to be more DRY (Don't Repeat Yourself) by creating reusable helper functions.

Action I can take to improve my skills and learning (make these “SMART”):

Add Sequelize Model Validations by 20 Sept

Implement comprehensive validation checks for required fields and proper data formats within Sequelize models to avoid bad data in the database.

Refactor Fee Controller Logic by 22 Sept

Refactor the fee management logic to remove repetitive code and improve reusability by creating helper functions.

LEARNING LOG 8

For the period 13/09/2024 to 24/09/2024

What have I done? (relate tasks to Learning Outcomes)

LO1: Effective Time Management

Finalized all backend modules and focused on documentation, deployment configuration, and code optimization before handover.

LO2: Teamwork and Collaboration

Worked with supervisor to prepare .env, database credentials, and deployment scripts for production.

LO3: Adaptability

Adapted the system to support environment-based configuration using dotenv. Separated DB configs between development and production.

LO4: Problem-Solving Skills

Identified and fixed broken token verification in the production environment. Added fallback logging and improved error messages in controllers.

LO5: Communication Skills

Delivered a complete walkthrough of routes, models, and controller logic to the frontend team and supervisor.

LO6: Node.js Backend Development

- Finalized all routes in Express, ensuring all point to controller methods only
- Cleaned up controllers to return consistent response shapes
- Removed test routes and logs before deployment

LO7: Express.js Development

- Implemented global error handler in middleware/error.js
- Applied try/catch around all async controller methods
- Ensured CORS and JSON body parsing are configured in app.js

LO9: Version Control with Git

- Final merge of all modules into main

What I did well (Refer to skills used)

Final Integration and Deployment Prep

Finalized all backend routes and ensured error handling was centralized. Cleaned up logs and removed test artifacts for production readiness.

Environment Configuration & Database Seeding

Used .env for environment-based setups and ran final Sequelize migrations. Seeded essential data (roles, fee types) and verified database constraints.

Cross-Department Handoff and Explanation

Collaborated with the Frontend Team and supervisor to complete the deployment setup. Provided a walkthrough to the supervisor on how endpoints operate post-deployment.

What I could improve on (i.e. skills I want to improve)

Post-Deployment Health Checks

While the backend is now production-ready, I need to implement health check routes to monitor the health of the application and its dependencies post-deployment.

Consistent API Response Formatting

There’s still room for improvement in ensuring consistency across all API responses. Some controllers return raw database models, and others use custom response formats. This can be standardized for better consistency.

Action I can take to improve my skills and learning (make these “SMART”):

Set Up Health Check Route

Develop a /health route to monitor application and database health after deployment, ensuring quick identification of issues in production.

Research Database Monitoring Tools

Investigate and implement tools such as PM2 or New Relic to monitor database performance and resource usage to keep the backend optimized.

Standardize API Responses

Create a consistent format for API responses to ensure uniformity across the backend. This can be achieved by using response wrappers or standardizing controller responses.