**Weekly Progress Report**

Name: Sahil Mishra Devendra

Domain: Full Stack Web Development

Date of submission: 16/07/2025

Week Ending: 02

# I. Overview:

This week, I focused on **two core** areas: Firstly, I set up the **complete development environment** for the **Multi-Client Website for Services project**. Secondly, I strengthened my **understanding of Java OOPs concepts**, revising key topics like Inheritance, Polymorphism, Encapsulation, and Abstraction, which will support backend design patterns during this project.

# II. Achievements:

**1. Java Learning and Revision:**

- Revised core OOPs concepts including Inheritance (single & multilevel), Super and Final keywords.  
- Practiced Abstraction using abstract classes and interfaces.  
- Reinforced Encapsulation by using private fields with public getters/setters.  
- Implemented Polymorphism with method overloading and overriding.  
- Studied constructors (default, parameterized, copy) and applied them in sample Java programs.

**2. Project Environment Setup:**

- Installed and configured Node.js, npm, React, and PostgreSQL locally.  
- Created Dockerfile and docker-compose.yml for containerizing backend API and database.  
- Initialized GitHub repository with main and dev branches, added .gitignore and README.  
- Configured Prettier, ESLint, and Husky pre-commit hooks for code consistency.  
- Bootstrapped Express.js server with test routes and integrated Sequelize ORM with PostgreSQL.  
- Set up base React project with React Router and Axios for API calls.

**3. Backend Kickoff:**

- Bootstrapped Express.js server with a working /health route for testing.  
  
- Integrated Sequelize ORM, connected to PostgreSQL, and created draft models for User, Client, and Service.  
- Created .env file for local config (JWT secret, DB URL, ports).

**4. Frontend Kickoff:**

- Set up base React structure with React Router.  
- Designed minimal wireframe for login page & client dashboard.  
- Installed Axios for connecting with backend APIs.

**5. CI/CD Initial Work:**

- Explored GitHub Actions for automating linting and testing workflows.  
- Drafted a workflow file for future deployment pipelines.  
- Studied deploying containerized applications on AWS EC2.

# III. Challenges

- Clarified how Java handles hybrid and multiple inheritance via interfaces.  
- Faced Docker networking issues connecting containers, resolved through Docker Compose tweaks.  
- Balanced time between theoretical Java practice and practical environment setup.

# IV. Learning Resources

- Java OOPs PPT and Oracle Documentation  
- Docker official tutorials and community examples  
- Sequelize and Express.js guides  
- GitHub Actions and AWS deployment tutorials

# V. Next Week's Goals

- Implement JWT-based signup/login APIs and test with Postman.  
- Build CRUD endpoints for services management.  
- Connect frontend forms with backend APIs using Axios.  
- Write unit tests for authentication logic using Jest.  
- Finalize database schema design for multi-tenancy.  
- Begin writing API documentation (Swagger/OpenAPI).

# VI. Additional Comments

By combining Java theory with practical environment setup, I am building a strong base for clean, modular backend code. This week’s setup will make the development workflow smoother and ready for upcoming features.

# VII. Lessons Learned

Revising Java OOPs improved my understanding of core principles that shape robust backend architectures. Setting up Docker early showed me the importance of containerization for consistent environments. I also improved my familiarity with CI/CD basics, which will help automate future deployments.