Contract Monthly Claim System

The Contract Monthly Claim System (CMCS) is a web application developed to simplify the process of submitting and approving monthly claims for contract teachers. For this project, I chose to use ASP.NET Core MVC on the backend and Nuxt.js / Vue.js on the frontend. This combination allows for a clear separation of concerns, with MVC providing a robust framework for business logic and Vue.js providing a dynamic and reactive interface. The initial prototype focuses on design and usability, without full functionality, which will be implemented in subsequent parts.

Design Choices

1. ASP.NET Core MVC: Chosen for its ability to organize code into models, views, and controllers, facilitating maintenance and scalability. Furthermore, it integrates seamlessly with the .NET ecosystem.

2. Vue.js: Chosen for the front-end due to its reactive, component-based nature, which enhances the user experience with dynamic updates without page reloads. Vue.js is easy to integrate with ASP.NET Core via CDN for prototyping.

3. Entity Framework Core: Designed for data access in the functional phase, using a code-first approach to map classes to the database.

Database Structure

The database will be relational (SQL Server) with the following main entities:

1. Lecturer: Stores teacher information (ID, name, email).

2. Claim: Stores submitted complaints (ID, hours worked, hourly rate, total amount, status, submission date, teacher ID).

3. Document: Stores supporting documents (ID, file name, content type, complaint ID).

Relationships:

1. A Teacher can have multiple Complaints (1 to N).
2. A Complaint can have multiple Documents (1 to N).

GUI Layout

The interface was designed to be intuitive and efficient:

Login Page: For authentication of all users

Teacher View: Dashboard and submission form

Coordinator/Manager View: List of pending complaints

Assumptions and Constraints

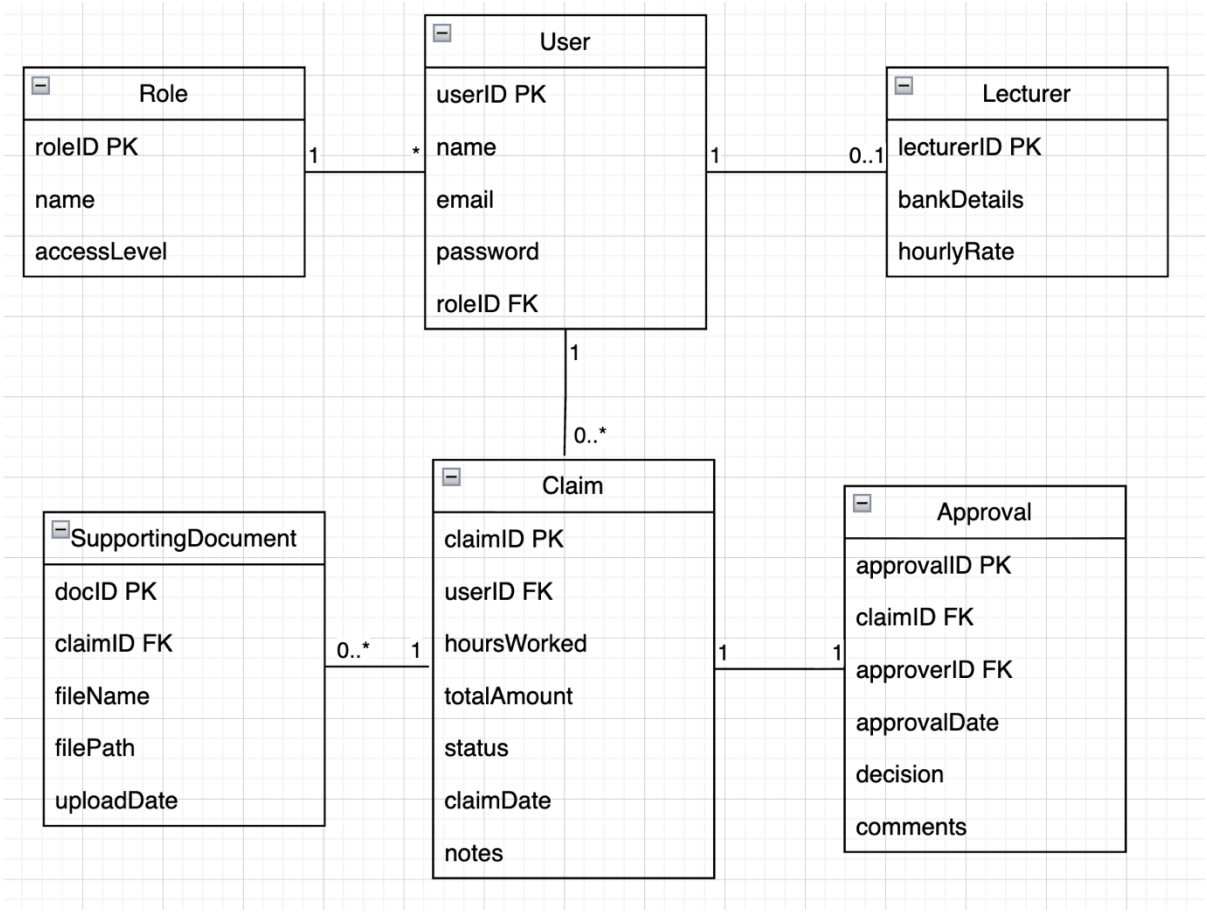
Assumptions:

1. Each teacher has a unique email address.
2. Claims are submitted monthly.
3. Coordinators and managers have role-based access.

Constraints:

1. The prototype is non-functional; there is no data persistence.
2. Document uploads are simulated; there is no real storage.
3. Authentication is not implemented in Part 1.

UML Class Diagram

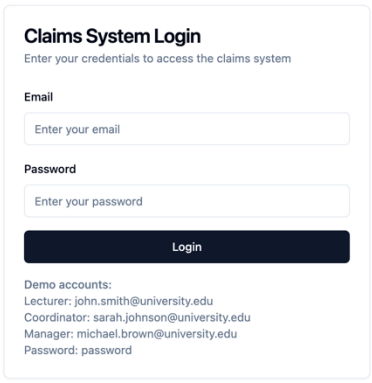


Project Plan

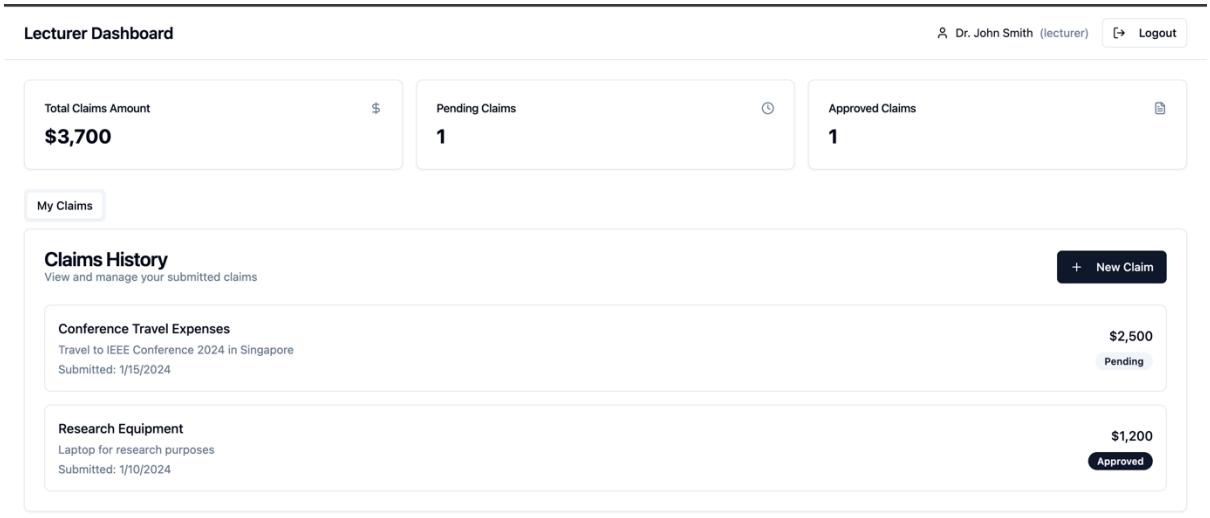
Task	Start Date	End Date	Dependency	Hours
Requirement Analysis	2025-10-01	2025-10-03	None	5
Complete UML Design	2025-10-04	2025-10-05	Requirements	4
Project Setup (MVC + Vue.js)	2025-10-06	2025-10-06	UML	2
Model Creation	2025-10-07	2025-10-07	Project Setup	3
Controller Creation	2025-10-08	2025-10-08	Models	4
View Creation (All Pages)	2025-10-09	2025-10-12	Controllers	8
Integration with Vue.js	2025-10-13	2025-10-13	Views	4
Documentation Writing	2025-10-14	2025-10-15	All Tasks	4
Review and GitHub Commit	2025-10-16	2025-10-15	Documentation	4
Total				38

GUI/UI

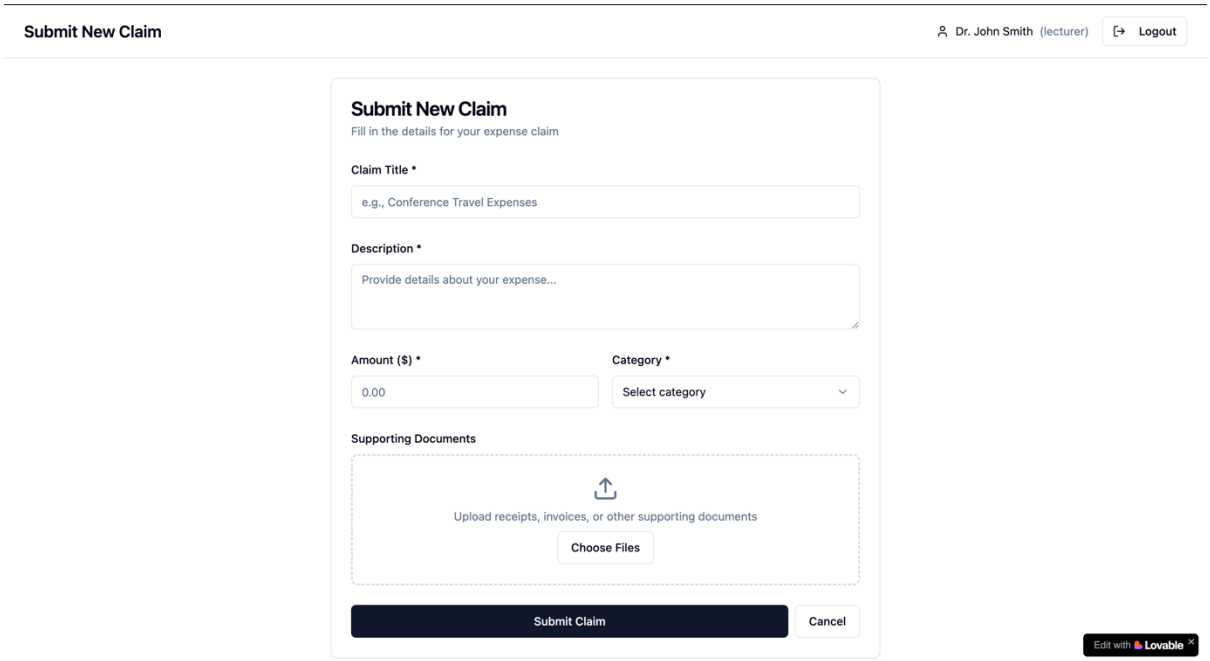
Login



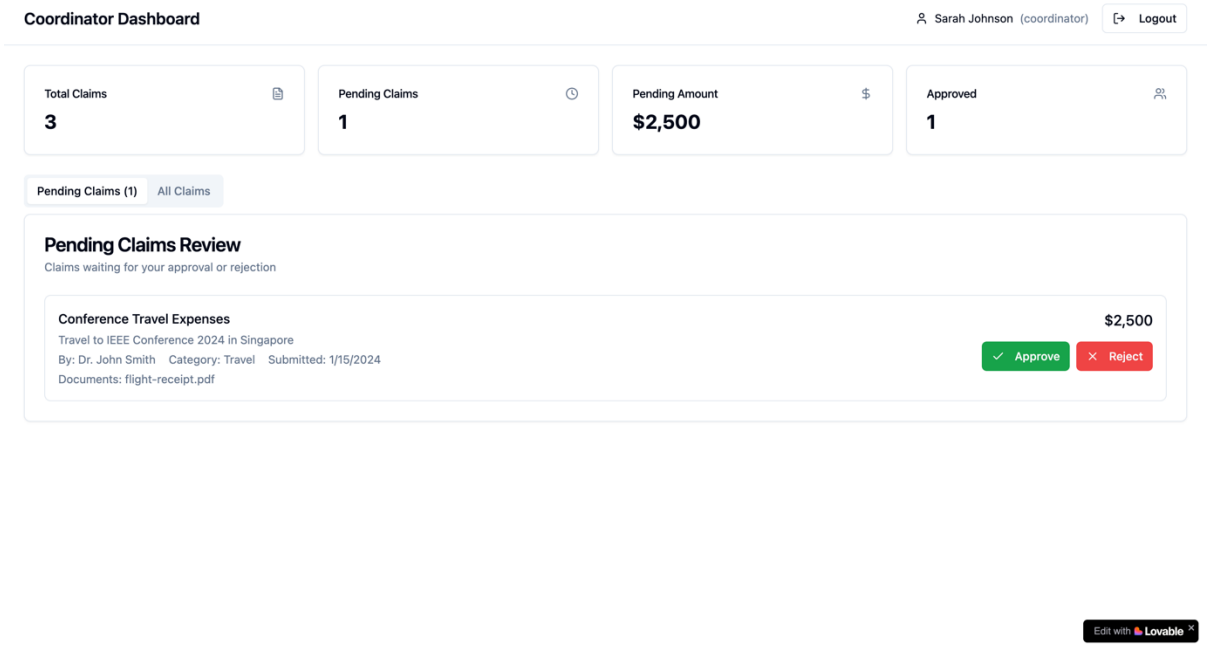
Lecturer Dashboard



Lecturer Form



Coordinator View



Coordinator Dashboard

Sarah Johnson (coordinator) Logout

Total Claims	Pending Claims	Pending Amount	Approved
3	0	$0	2

Pending Claims (0) | All Claims

All Claims

Complete history of all claims in the system

Conference Travel Expenses — $2,500 — Approved
Travel to IEEE Conference 2024 in Singapore
By: Dr. John Smith Category: Travel Submitted: 1/15/2024 Reviewed: 9/8/2025

Research Equipment — $1,200 — Approved
Laptop for research purposes
By: Dr. John Smith Category: Equipment Submitted: 1/10/2024 Reviewed: 1/12/2024

Workshop Materials — $350 — Rejected
Materials for student workshop
By: Prof. Alice Wilson Category: Supplies Submitted: 1/8/2024 Reviewed: 1/9/2024