uCU INNOVATORS HUB PROJECT REPORT

**TABLE OF CONTENTS**

1. Introduction
2. Problem Statement
3. Objectives
4. System Requirements
5. System Design
6. System Architecture Diagram
7. Methodology
8. System Implementation
9. Testing & Troubleshooting
10. Challenges and Solutions
11. Results
12. Conclusion
13. Future Work

## 1. Introduction

This report presents the development, implementation, testing, and deployment of the UCU Innovators Hub system. The system is a full‑stack web application designed to manage innovation projects, enable authentication, and support CRUD operations for project data. The project follows modern software engineering principles and addresses both functional and technical requirements.

## 2. Problem Statement

Students and innovators at UCU lacked a digital platform to submit, track, and manage innovation‑related projects. Lecturers also needed a centralised system for evaluating and monitoring student projects. The absence of such a platform created challenges in accessibility, collaboration, and project documentation. The UCU Innovators Hub was built to solve these issues by providing a digital platform with user authentication, project management, and administrative controls.

## 3. Objectives

- Develop a functional full‑stack platform for managing innovation projects.

- Implement user authentication using JWT.

- Provide CRUD operations for projects.

- Build a clean UI using React and Bootstrap.

- Deploy a working prototype suitable for demonstration.

## 4. System Architecture

The platform uses a client–server architecture following MVC patterns on the backend.

Frontend: React + Axios + Bootstrap

Backend: Node.js + Express.js

Database: MongoDB

Authentication: JWT-based authentication using access tokens stored in browser storage.

Communication: RESTful API endpoints.

## 5. Technologies Used

- React (Frontend UI)

- Axios (HTTP Requests)

- Bootstrap (Styling)

- Node.js & Express.js (Backend)

- MongoDB & Mongoose (Database)

- JWT (Authentication)

- Postman (API Testing)

- GitHub (Version Control)

## 6. Challenges & How They Were Solved

1. Frontend failing due to jwt-decode import error . We resolved this by updating the import syntax to `import { jwtDecode } from 'jwt-decode';` since version 4 no longer supports default imports.

2. Axios not attaching authorization tokens. Solved by adding Axios interceptors in index.js to automatically attach the Bearer token to headers.

3. Projects not loading (401 Unauthorized). Fixed by ensuring the backend verifies tokens correctly and returns proper JSON responses.

4. Node modules corruption. Solved by deleting node\_modules and package-lock.json then reinstalling using `npm install`.

## 7. System Modules

- Authentication Module (Login, Token Handling)

- User Dashboard

- Projects Module (Add, Update, Delete, View)

- Admin Features (Future Work)

- API Integration Layer

## 8. Testing Report

Testing was performed using both manual and automated approaches.

1. API Testing (Postman)

- Endpoints tested: /login, /projects, /projects/:id

- Status codes verified: 200, 201, 400, 401, 404

2. Frontend Testing

- Component rendering

- Form input validation

- Token handling logic

- Error boundary handling

3. Integration Testing

- React–Backend communication

- JWT token validation

- Project CRUD operations

## 9. Deployment Guide

Frontend:

- Install dependencies using `npm install`.

- Start development server using `npm start`.

Backend:

- Run `npm install`.

- Configure environment variables (.env).

- Start backend using `npm run dev`.

Database:

- Use MongoDB Atlas or Local MongoDB instance.

- Create collections automatically on first API use.

## 10. Conclusion

The UCU Innovators Hub successfully demonstrates the application of full‑stack development principles. It delivers authentication, project management, and a functional API layer. With additional development, it can scale into a fully‑deployed innovation management ecosystem for the university.

## 11. References (IEEE Style)

> M. Grinberg, \*Flask Web Development\*, O'Reilly Media, 2018.

> Express.js, 'Express — Node.js web application framework', 2024. [Online]. Available: https://expressjs.com.

> ReactJS, 'React Documentation', 2024. [Online]. Available: https://react.dev.

> MongoDB, 'MongoDB Documentation', 2024. [Online]. Available: https://mongodb.com.

WRITTEN BY,

**NAKAAYI SHAMIRAH**

**NALULE ESTHER PRECIOUS**