# Outfithub — Project Report

## Topic

**E-commerce Web Application**

## Title

**Outfithub — Modular Clothing E-commerce Application**

## Author / Owner

**Seng-Chaokhun** Repository: *Outfithub*

## Date

**January 15, 2026**

---

## Team Members

| Member | ID | Responsibilities |
|---|---|---|
| Seng Chaokhun | e20220478 | Project Manager |
| Thou Laiheng | e20220478 | QA & Tester |
| Ang Panha | e20221707 | Code Review & Architecture |
| Kham Veasna | e20220478 | QA & Tester |
| Cheng Sakda | e20220190 | Report Writing |

---

## Executive Summary

Outfithub is a modular e-commerce web application focused on online clothing sales. The project is built using **Vue 3 with TypeScript** and bundled with **Vite**, emphasizing clean architecture, component reusability, and maintainability.

The application supports **user authentication**, **product browsing**, and a structured frontend–backend interaction using RESTful APIs. A Node.js + Express backend and MySQL database are included within the same workspace, allowing tighter integration during development. State management is handled with **Pinia**, while tooling such as ESLint, Prettier, and Vitest ensures code quality and consistency.

---

## Project Purpose and Scope

### Purpose

The purpose of Outfithub is to provide a **simple, user-friendly online clothing store** that demonstrates modern frontend practices, authentication workflows, and modular project structure suitable for academic and real-world development.

### Scope Implemented

- User authentication (register, login, logout)
- Product listing and browsing
- Clothing-focused catalog presentation
- Modular UI components
- Frontend–backend integration via REST APIs
- Centralized state management for authentication and user data

---

## Technology Stack

### Frontend

- **Framework:** Vue 3
- **Language:** TypeScript
- **Build Tool:** Vite
- **Routing:** Vue Router v4
- **State Management:** Pinia
- **Styling:** Tailwind CSS, Sass
- **Icons:** Lucide Vue Next

### Backend

- **Runtime:** Node.js
- **Framework:** Express (v5)
- **Authentication:** Custom auth logic with encrypted passwords
- **Password Hashing:** bcryptjs
- **Environment Config:** dotenv
- **CORS Handling:** cors

### Database

- **MySQL** (via mysql2)

### Tooling & Quality

- **Linting:** ESLint
- **Formatting:** Prettier
- **Type Checking:** vue-tsc
- **Testing:** Vitest
- **Dev Utilities:** Nodemon, tsx
- **Script Orchestration:** npm-run-all2

---

## Project Architecture

Outfithub follows a **modular, layered architecture** with a clear separation of concerns.

### High-Level Structure

```
/
├── src/                # Vue frontend
│   ├── components/     # Reusable UI components
│   ├── pages/          # Page-level views
│   ├── layouts/        # Layout wrappers (auth, user, admin)
│   ├── router/         # Route definitions
│   ├── stores/         # Pinia state stores
│   ├── assets/         # Global styles and assets
│   └── main.ts         # App entry point
│
├── server/             # Express backend
│   ├── routes/         # API routes
│   ├── controllers/   # Request handling logic
│   ├── config/         # Database and env config
│   └── index.ts        # Server entry point
│
├── public/             # Static assets
└── config files        # Vite, TS, ESLint, Vitest configs
```

### Architectural Rationale

- **Frontend and backend are colocated** to simplify development and deployment.
- **Layouts** enforce role-based UI separation.
- **Pinia stores** centralize business logic and API communication.
- **Component-driven design** improves reusability and consistency.

---

## Authentication System

The application includes a complete authentication flow:

- User registration with encrypted password storage
- User login and logout
- Authentication state managed globally using Pinia
- Axios used for secure communication with backend endpoints
- Environment variables used to protect sensitive configuration

Passwords are hashed using **bcryptjs**, and authentication logic is isolated from UI components to improve security and maintainability.

---

## Features Implemented

- Clothing product listing
- Modular product card components

- Authentication pages (login & register)
- Auth-protected routes
- Centralized auth state management
- Reusable layout system
- Responsive UI using utility-based styling
- Backend API for user and product data

---

## Developer Scripts

Key npm scripts used in development:

- **Development**

  ```
  npm run dev
  ```

- **Backend server**

  ```
  npm run server
  ```

- **Production build**

  ```
  npm run build
  ```

- **Type checking**

  ```
  npm run type-check
  ```

- **Lint & format**

  ```
  npm run lint
  npm run format
  ```

- **Unit testing**

  ```
  npm run test:unit
  ```

---

## Development Process

1. **Project setup** using Vite + Vue 3 + TypeScript
2. **Backend initialization** with Express and MySQL
3. **Authentication system implementation**
4. **Routing and layout separation**
5. **Component-based UI construction**
6. **State management with Pinia**
7. **Code quality tooling integration**
8. **Testing and build preparation**

---

## Testing and Code Quality

- ESLint and Prettier enforce consistent coding standards
- Type safety ensured through vue-tsc
- Vitest configured for unit testing
- Modular structure enables isolated testing of components and stores

---

## UI/UX Considerations

- Clean and minimal interface suitable for e-commerce
- Clear navigation between authentication and shopping areas
- Consistent visual hierarchy across pages
- Reduced user friction during login and browsing
- Responsive layout optimized for desktop and tablet use

---

## Limitations

- No payment gateway integration
- Limited product management features
- No order history or checkout system
- Accessibility testing not fully implemented

---

## Future Improvements

- Shopping cart and checkout workflow

- Payment integration (Stripe or PayPal)

- Admin product management dashboard

- Improved mobile responsiveness

- Accessibility (ARIA labels, keyboard navigation)

- Shopping cart and checkout workflow

- Payment integration (Stripe or PayPal)

- Admin product management dashboard
- Improved mobile responsiveness
- Accessibility (ARIA labels, keyboard navigation)
- End-to-end testing and CI pipeline

---

# Conclusion

Outfithub demonstrates a solid implementation of a modern Vue-based e-commerce application with authentication and modular architecture. The project emphasizes maintainability, separation of concerns, and real-world development practices, making it suitable as both an academic submission and a foundation for future expansion.