
CIT 222 - Information System Analysis and Design

Sessional Project Report

Project Title : Board Autoprofiler

Last Updated on : Wed 28, May 2025

Submission Date : Wed 28, May 2025

Submitted to,

Md Atikur Rahman

Professor,

Department of Computer Science and Information Technology,
Patuakhali Science and Technology University.

Submitted by,

1. **Md. Sadman Kabir Bhuiyan**

ID : 2102020,

Reg: 10147

2. **Md. Sharafat Karim**

ID : 2102024,

Reg: 10151

Contents

1. Introduction	3
2. Objective	3
3. Scopes	3
4. Problem Statement	3
4.1. Target Audience	3
5. Requirements	3
5.1. Functional Requirements	3
5.2. Non-Functional Requirements	3
6. Technology	4
7. Architecture	4
7.1. Overview	4
7.2. MACH Architecture	4
7.3. MVC Integration	4
7.4. E-R Diagram	5
8. Implementation	5
8.1. Microservices Architecture	5
9. Conclusion	6
10. References	6
10.1. Feasibility study	6
10.2. Documentations	6

Board Autoprofiler

1. Introduction

The Board Autoprofiler is a web-based system designed to simplify the process of selecting compatible PC components. It leverages intelligent recommendation algorithms to ensure optimal performance, compatibility, and benchmarking. The system also integrates e-commerce functionality, enabling users to purchase recommended components directly.

Online Store URL : <https://demo.board-autoprofiler.sharafat.xyz/>
Admin URL : <http://board-autoprofiler.runasp.net>
Business URL : <https://board-autoprofiler.sharafat.xyz>

2. Objective

To develop a web-based system that provides intelligent recommendations for computer hardware components based on user selections, ensuring compatibility, performance optimization (via bottle-neck analysis), and benchmarking. Additionally, the system will include an e-commerce portal for customers and an admin dashboard for managing products and data.

3. Scopes

The project scope includes the development of a recommendation engine, e-commerce portal, and admin dashboard. It aims to address compatibility issues, performance bottlenecks, and benchmarking needs for PC components.

4. Problem Statement

Currently, users face difficulties when selecting compatible and balanced PC components. Existing platforms like Ryans Computers Limited allow purchasing but lack intelligent recommendation systems that consider performance metrics, bottlenecks, and benchmarks.

4.1. Target Audience

- General users looking to build or upgrade their PCs.
- Computer repair shops and technicians.
- Gamers seeking optimal performance.
- Professionals needing specific hardware configurations.

5. Requirements

5.1. Functional Requirements

1. User selects PC components (e.g., CPU, GPU, RAM).
2. System recommends compatible parts based on selection.
3. Performance bottleneck detection and suggestions.
4. Benchmark comparison between selected and recommended components.
5. E-commerce functionality: product browsing, cart, checkout.
6. Admin panel to manage inventory, add/edit products.
7. User registration and login.
8. Search and filter functionality.

5.2. Non-Functional Requirements

1. System should be responsive and user-friendly.

2. Fast response time for recommendations.
3. Secure user data and transactions.
4. Scalable to handle multiple users and products.

6. Technology

Layer	Technology
Frontend	HTML/CSS, Bootstrap, JavaScript, React.js (Optional)
Backend	ASP.NET MVC Core (.NET 8+)
Microservices	ASP.NET Web API
Database	Microsoft SQL Server
ORM	Entity Framework Core
Authentication	JWT / ASP.NET Identity
Hosting	MonsterASP.net / Azure / AWS / Local IIS
CI/CD	GitHub Actions

7. Architecture

7.1. Overview

The system uses a microservices architecture with MVC pattern integration to ensure modularity, scalability, and maintainability. The Recommendation Microservice and E-commerce Microservice are independently deployable, communicating via RESTful APIs through an API Gateway. The MVC pattern structures the application logic within each microservice and the frontend, separating data (Model), presentation (View), and business logic (Controller).

7.2. MACH Architecture

The architecture follows MACH principles (Microservices, API-first, Cloud-native, Headless) and is divided into:

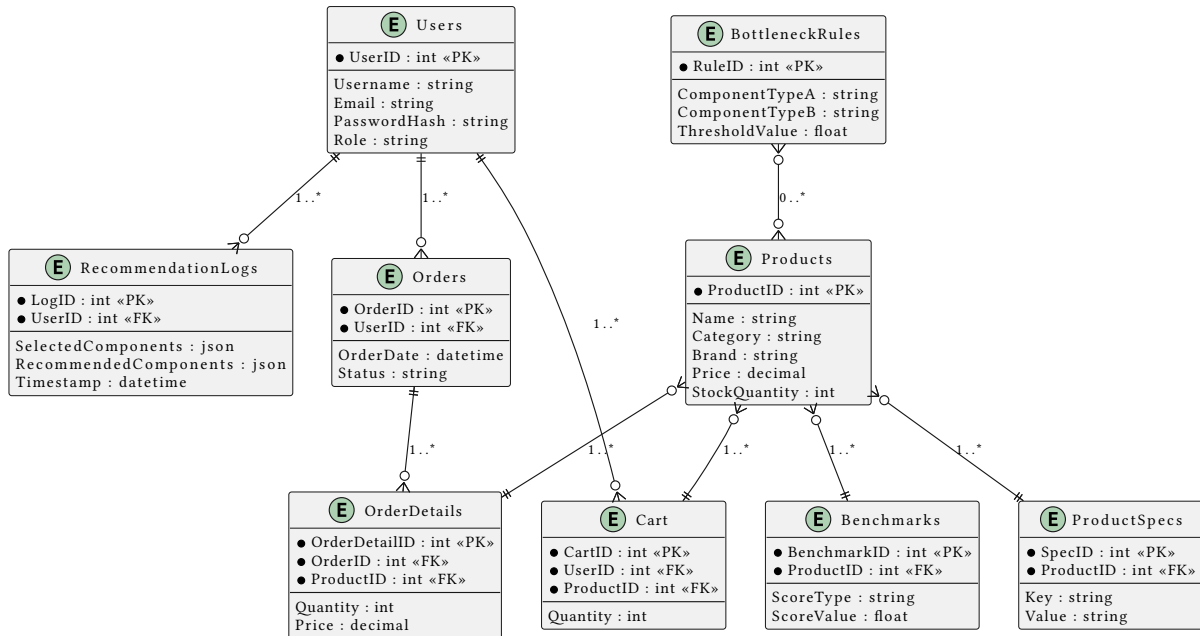
1. **Client Tier:** Web and mobile interfaces built with React.js, following MVC for frontend logic.
2. **Application Tier:** Microservices with MVC-based backend logic (Node.js for E-commerce, Python for Recommendation).
3. **Data Tier:** Distributed databases tailored to each microservice's needs.

7.3. MVC Integration

1. **Model:** Represents data and business logic.
 - In the Recommendation Microservice, the Model includes component specifications, compatibility rules, and benchmark data stored in MongoDB and Elasticsearch.
 - In the E-commerce Microservice, the Model includes product catalog, cart state, and order data stored in MongoDB, PostgreSQL, and Redis.
2. **View:** Handles presentation and user interface.
 - Built with React.js and Tailwind CSS, rendering dynamic UI components for product listings, recommendation results, and admin dashboards.
 - Views fetch data from Controllers via API calls and update based on user interactions.
3. **Controller:** Manages communication between Model and View.

- In microservices, Controllers are RESTful API endpoints (Node.js for E-commerce, Python/Flask for Recommendation) that process requests, query Models, and return data to Views.
- Frontend Controllers in React.js handle user events (e.g., component selection, checkout) and update Views accordingly.

7.4. E-R Diagram



8. Implementation

8.1. Microservices Architecture

1. Component Recommendation Service

- API Endpoints:
 - /api/recommendations
 - POST: Accepts selected components
 - Returns: Compatible components + bottleneck analysis
- Logic:
 - Parse selected components
 - Query database for specifications and benchmarks
 - Apply rules to detect bottlenecks
 - Recommend compatible alternatives

2. E-Commerce & Admin Service

- API Endpoints:
 - /api/products
 - /api/cart
 - /api/orders
- Features:
 - Customer can browse and purchase
 - Admin can manage products, stock, categories

9. Conclusion

Board Autoprofiler aims to revolutionize the way users select PC components by combining intelligent recommendation systems with real-time performance analysis. By addressing the gaps in existing platforms like Ryans Computers, this system empowers users to make informed decisions while building or upgrading their desktop PCs. With a scalable microservices architecture, robust security, and user-friendly interface, the system is well-positioned for long-term success and expansion.

10. References

10.1. Feasibility study

- <https://pc-builds.com/>
- <https://pc-builds.com/bottleneck-calculator/>
- <https://www.ryans.com/>
- <https://www.startech.com.bd/>

10.2. Documentations

- <https://www.w3schools.com/html/> [**W3Schools HTML**]
- <https://www.w3schools.com/css/> [**W3Schools CSS**]
- <https://www.w3schools.com/js/> [**W3Schools JavaScript**]
- <https://www.w3schools.com/bootstrap5/index.php> [**W3Schools Bootstrap 5**]
- <https://reactjs.org/docs/getting-started.html> [**React.js docs**]
- <https://tailwindcss.com/docs> [**Tailwind CSS docs**]
- <https://docs.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0> [**ASP.NET Core docs**]

THE END