

**Pc and More**

by

**Mohammad Shaaban**

**Mohamed Farhat**

**Mohammad Alhalim**

**Ali hneineh**

**Houssein Harb**

**Project**

Submitted in Partial Fulfillment of the Requirements for the Degree of bachelor’s in computer engineering

Department of Engineering

Faculty of Engineering

Year Spring- 2025



**Pc and More**

by

**Mohammad Shaaban**

**Mohammed Farhat**

**Mohammad Alhalim**

**Ali Hneineh**

**Houssein Harb**

**Project**

Submitted in Partial Fulfillment of the Requirements for the Degree of bachelor’s in computer engineering

Department of Engineering

Faculty of Engineering

**Supervised by**

**Prof. /Dr Imane Haidar**

**Contents**

[List of Figures 6](#_Toc194762447)

[List of Tables 7](#_Toc194762448)

[Project Description 8](#_Toc194762449)

[Project Overview 8](#_Toc194762450)

[Objectives 8](#_Toc194762451)

[Background 8](#_Toc194762452)

[Literature Review 8](#_Toc194762453)

[PCPartPicker (USA) 8](#_Toc194762454)

[Newegg (USA) 9](#_Toc194762455)

[PC and Parts (Lebanon) 9](#_Toc194762456)

[Research Paper: AI-based PC Build Optimization (2021) 9](#_Toc194762457)

[Research Paper: AI Chatbots in E-commerce (2020) 9](#_Toc194762458)

[Summary Table 10](#_Toc194762459)

[Applications 11](#_Toc194762460)

[Online PC Retail 11](#_Toc194762461)

[Computer Retail Stores 11](#_Toc194762462)

[Alternative Designs 11](#_Toc194762463)

[Shopify 11](#_Toc194762464)

[Magento (Adobe Commerce) 11](#_Toc194762465)

[Desktop application 11](#_Toc194762466)

[Google Cloud Vision API 11](#_Toc194762467)

[Custom OpenCV + TensorFlow Model 11](#_Toc194762468)

[WhatsApp Chatbot 11](#_Toc194762469)

[Our Design 11](#_Toc194762470)

[Project Planning 12](#_Toc194762471)

[Constraints 12](#_Toc194762472)

[Implementation environment of the current system 12](#_Toc194762473)

[Partners and collaboration 12](#_Toc194762474)

[Off-the-shelf Software 12](#_Toc194762475)

[Schedule Constraints 12](#_Toc194762476)

[Budget Constrains 12](#_Toc194762477)

[Project Issues 13](#_Toc194762478)

[Data Security Implementation Strategy 13](#_Toc194762479)

[Supplier and Inventory Management System 13](#_Toc194762480)

[Data Migration Process 13](#_Toc194762481)

[AI Model Bias and Inaccuracy 13](#_Toc194762482)

[Financial Sustainability 13](#_Toc194762483)

[Team Members Tasks 14](#_Toc194762484)

[AI/ML Developers 14](#_Toc194762485)

[Web Developers 14](#_Toc194762486)

[Data Collection 14](#_Toc194762487)

[Work Documentation(Report and README) 14](#_Toc194762488)

[Other Development Tools 14](#_Toc194762489)

[Ethical Issues 14](#_Toc194762490)

[1. Data Privacy and Security 14](#_Toc194762491)

[2. Transparency in AI Decision-Making 14](#_Toc194762492)

[3. Bias and Fairness in AI Models 14](#_Toc194762493)

[4. Intellectual Property Rights 14](#_Toc194762494)

[Software Model Process 15](#_Toc194762495)

[Waterfall model 15](#_Toc194762496)

[Reuse-Oriented Approuch 15](#_Toc194762497)

[Feasibility Study 16](#_Toc194762498)

[Demand for PC Parts 16](#_Toc194762499)

[Competitor Analysis 16](#_Toc194762500)

[AI Model Feasibility 16](#_Toc194762501)

[Estimated Costs 16](#_Toc194762502)

[Break-even Analysis 16](#_Toc194762503)

[Feasibility Verdict: Proceed with Adjustments 16](#_Toc194762504)

[Tools/Technology 17](#_Toc194762505)

[Programming Languages 17](#_Toc194762506)

[FrameWorks and Libraries 17](#_Toc194762507)

[API Keys 17](#_Toc194762508)

[Development Tools 17](#_Toc194762509)

[Standards 17](#_Toc194762510)

[IEEE 23026-2015 – Engineering and Management of Websites for Systems, Software, and Services Information. 17](#_Toc194762511)

[UX/UI Design Conventions 17](#_Toc194762512)

[Performance & SEO Conventions 17](#_Toc194762513)

[Milestones 18](#_Toc194762514)

[Project Kickoff & Requirement Gаthеring (✅ Comрlеtеd) 18](#_Toc194762515)

[System Arсhitесturе & Tесhnology Stасk Finаlizаtion (✅ Comрlеtеd) 18](#_Toc194762516)

[Prototyре Dеvеloрmеnt (✅ Comрlеtеd) 18](#_Toc194762517)

[AI Configurаtor & Hаrdwаrе Rесognition Model Imрlemеntаtion (⏳ In Progrеss) 18](#_Toc194762518)

[Bеtа Tеsting & Usеr Fееdbасk Collесtion (🔜 Uрсoming) 18](#_Toc194762519)

[Pеrformаnсе Oрtimizаtion & Sсаlаbility Tеsting (🔜 Uрсoming) 18](#_Toc194762520)

[Finаl Dерloymеnt & Mаrkеting Lаunсh (🔜 Uрсoming) 18](#_Toc194762521)

[Requirements 19](#_Toc194762522)

[Use Cases 19](#_Toc194762523)

[Functional Requirements 19](#_Toc194762524)

[User Registration and Login 19](#_Toc194762525)

[Product Catalog Management 19](#_Toc194762526)

[Search and Product Recognition 19](#_Toc194762527)

[AI-Powered PC Configurator 19](#_Toc194762528)

[Chatbot Integration 19](#_Toc194762529)

[Shopping Cart and Checkout 19](#_Toc194762530)

[Order Management 19](#_Toc194762531)

[Payment Gateway 19](#_Toc194762532)

[Notifications 19](#_Toc194762533)

[Data Requirements 20](#_Toc194762534)

[Product Data 20](#_Toc194762535)

[User Data 20](#_Toc194762536)

[Order Data 20](#_Toc194762537)

[AI Model Data 21](#_Toc194762538)

[Payment Data 21](#_Toc194762539)

[Review & Rating Data 21](#_Toc194762540)

[Inventory & Supplier Data 21](#_Toc194762541)

[Non-Functional Requirements 22](#_Toc194762542)

[Performance Requirements 22](#_Toc194762543)

[Scalability Requirements 22](#_Toc194762544)

[Availability Requirements 22](#_Toc194762545)

[Security Requirements 22](#_Toc194762546)

[Usability Requirements 22](#_Toc194762547)

[Reliability Requirements 22](#_Toc194762548)

[Portability Requirements 22](#_Toc194762549)

[Legal Requirements 22](#_Toc194762550)

[Design 23](#_Toc194762551)

[Class Diagrams 23](#_Toc194762552)

[Dynamic Model 23](#_Toc194762553)

[Subsystem Decomposition 23](#_Toc194762554)

[Hardware / software mapping 23](#_Toc194762555)

[User Interface 23](#_Toc194762556)

[Test Plans 24](#_Toc194762557)

[Implementation 25](#_Toc194762558)

[Results Evaluation 26](#_Toc194762559)

[Conclusion 27](#_Toc194762560)

[Summary 27](#_Toc194762561)

[Novelty 27](#_Toc194762562)

[Integrity and Values 27](#_Toc194762563)

[Future Work 27](#_Toc194762564)

[References / Bibliography 28](#_Toc194762565)

# List of Figures

[Figure 1: Use Case Diagram 18](#_Toc195974204)

[Figure 2: Class Diagram 24](#_Toc195974205)

[Figure 3: Context Diagram 25](#_Toc195974206)

[Figure 4: Sequence Diagram 25](#_Toc195974207)

[Figure 5: Classses and Association Diagram 26](#_Toc195974208)

# 

# Project Description

## Project Overview

The PC Parts Shop Website is an advanced e-commerce platform designed for PC lovers who want to get their new computer hardware effectively. The website integrates a chatbot for customer support (it can answer any question related to PCs), additionally it has an AI-powered hardware recognition model for identifying PC components, and an advanced AI that assists customers in creating the optimal PC based on their needs and budget.

## Objectives

* Develop a user-friendly website for buying PC components.
* Integrate a chatbot to assist customers with inquiries and questions.
* Implement a PC hardware recognition model for identifying components from images.
* Develop an AI-powered PC configurator to recommend compatible hardware setups.

## Background

The project lies under the e-commerce and technology domain, satisfying the growing demand regarding customized builds and PC hardware. As the uses of computers increases in multiple domains (AI applications, gaming, and content creation), the demands of better, more functional computers arise. However, people with no background information about computer hardware will face a problem trying to assemble one that meets their needs, here comes our software that enhances the users’ experience by providing for the user automated responses and recommendations through AI-powered chatbot assistance. The user might have a certain budget for the PC, using our AI integration, the user enters the budget and a complete build with the highest specs and detailed description about each hardware part is generated and everything is between the boundaries of the user's budget.

## Literature Review

The PC Parts Shop Website development requires research of existing products and related field studies as it features a chatbot alongside a PC hardware recognition model and a PC AI configurator. This literature review examines current web platforms alongside e-commerce sites and scholarly research to understand how they advance AI technology in PC component recommendations and customer service chatbots.

### PCPartPicker (USA)

1. Description: Online service for assembling and buying personalized PC setups. Offers automated compatibility checks along with price comparison tools and user reviews from community members.
2. Advantages:

* Ensures compatibility
* Vast product database
* Community-driven reviews

1. Problems:

* No chatbot integration
* Lacks AI-based recommendations

### Newegg (USA)

1. Description: Newegg operates as an online store that focuses exclusively on selling computer hardware components along with peripheral devices and additional accessories. Users can read customer reviews and product ratings while also finding frequent deals on the site.
2. Advantages:

* Large inventory
* Costumer review
* Discount combo deals

1. Problems:

* No AI-based PC builder
* Requires manual compatibility checking

### PC and Parts (Lebanon)

1. Description: This Lebanese online store provides computer components as well as laptops and various computer accessories. Supports both individual consumers and corporate clients.
2. Advantages:

* Supports local market
* Competitive pricing
* Customer support

1. Problems:

* No AI configurator
* Requires sending emails for PC configuration

### Research Paper: AI-based PC Build Optimization (2021)

1. Description: Discussion about implementing AI recommendation systems that can help users to optimize PC builds based on their budget, peripherals, and compatibility. For example, "I Tried Optimizing My PC With AI" is a YouTube video in which the creator explores AI to optimize a PC build.
2. Advantages:

* Improve cost-efficiency
* Ensure compatibility checks

1. Problems:

* Lack of real-time product availability tracking

### Research Paper: AI Chatbots in E-commerce (2020)

1. Description: Some Studies about chatbots in e-commerce with AI integration for better product selection and customer support.
2. Advantages:

* Reduces support workload
* Enhances user engagement
* Improves response accuracy

1. Problems:

* Struggles with technical queries
* Trust issues

### Summary Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Authors / Platform | Description | Advantages | Problems |
| 1 | PCPartPicker (USA) | Online platform for building and purchasing custom PC configurations. Provides automated compatibility checks, pricing comparisons, and community-driven reviews. | Ensure compatibility; Vast product database; Community-driven reviews | Lacks AI-based recommendations; No chatbot integration |
| 2 | Newegg (USA) | E-commerce site specializing in selling PC hardware, peripherals, and accessories. Provides customer reviews, product ratings, and frequent deals. | Large inventory; Customer reviews; Discount combo deals | No AI-based PC builder; Requires manual compatibility checking |
| 3 | PC and Parts (Lebanon) | Lebanese online store offering PC components, laptops, and accessories. Supports both individual consumers and corporate clients. | Supports local market; Competitive pricing; Customer support | No AI configurator; Limited online features |
| 4 | Research Paper: AI-based PC Build Optimization (2021) | Discusses the implementation of AI-driven recommendation systems to help users optimize PC builds based on budget, workload, and compatibility. | Improves cost-efficiency; Ensures compatibility checks | Lacks real-time product availability tracking |
| 5 | Research Paper: AI Chatbots in E-commerce (2020) | Study on AI-driven chatbots for product selection and customer support in e-commerce. | Reduces support workload; Enhances user engagement; Improves response accuracy | Struggles with technical queries; Trust issues |

Table 1:Summary Table

## Applications

### Online PC Retail

* Can be used as an online store for selling PC components, and accessories.
* Competes with websites like Newegg, PCPartPicker, and Amazon but it has an AI recommender.
* Helps Computer builders by suggesting compatible parts based on budget.

### Computer Retail Stores

* Can be used by physical Computer shops to improve customer experience.
* Employees can use the AI tools to recommend builds based on customer budget and needs.

## Alternative Designs

### Shopify

* A fully hosted, SaaS-based e-commerce platform that allows businesses to create and manage online stores with ease.

### Magento (Adobe Commerce)

* A powerful open-source e-commerce platform for large businesses.

### Desktop application

* A desktop-based e-commerce application could serve as a dedicated shopping platform with enhanced performance and offline capabilities. For example, we could integrate an offline AI chatbot.

### Google Cloud Vision API

* Pre-trained AI that detects objects, text, and product labels in images. Instead of using our gemini model.

### Custom OpenCV + TensorFlow Model

* If we want full control, we can train a custom TensorFlow or PyTorch model for product identification. This would be a better alternative and faster than our current model.

### WhatsApp Chatbot

* Instead of a website chatbot, we can integrate AI-powered support into WhatsApp or Messenger.

### Our Design

* What we did was use a A to Z website using the most used frameworks and integrated them with Gemini AI using API key (using python) and trained the model based on our dataset.

# Project Planning

## Constraints

### Implementation environment of the current system

* *Technology stack:* Our project is based on standard web technologies (HTML, CSS, JavaScript and some related frameworks, with a basic python backend), this environment is sufficient for only simple e-commerce site which limits our abilities in using more complex and efficient interface.

### Partners and collaboration

* *Technical background:* Our team has basic to intermediate knowledge in different technologies used, so this means that we cannot help each other when needed. If anything goes wrong in the backend for example, we won’t be able to help the backend dev guy.

### Off-the-shelf Software

* *Customization Limitations:* Using pre-built solutions speeds up development but may impose restrictions on customization and flexibility, potentially limiting the project’s uniqueness.

### Schedule Constraints

* *Fixed Academic Timeline:* The project must be finished within our semester, with main milestones (prototype, testing, final submission) that are scheduled throughout the term
* *Tight phases deadline:* Each phase has its own submission deadline (documentation submission); This means that going back to the old phase to add or modify any new feature would not work, we must stick with our pre-defined plan.

### Budget Constrains

* *Limited Funding:* Each team member contributes a little bit of his savings, so funding is minimal. This restricts spending on premium hosting services and purchasing the required APIs to achieve our project milestones.
* *Reliance on Free Resources:* The project primarily uses free and open-source solutions, which are cost-effective but might not offer the same performance as commercial alternatives.

## Project Issues

### Data Security Implementation Strategy

* The security protocols for user accounts (such as OAuth) compared to conventional email password authentication methods remain undecided.
* The evaluation of encryption techniques used for safeguarding sensitive user information is in debate, with Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithms being contemplated.
* The team has not yet reached a mutual consensus regarding the most effective strategy to ensure adherence to GDPR and PCI-DSS compliance requirements

### Supplier and Inventory Management System

* Delayed development of a real-time inventory update system.
* Unconcluded decisions on whether to build integration with supplier APIs or use manual entry for stock levels.
* There is a constraint concerning the management of out-of-stock items to avoid the risk of overselling.

### Data Migration Process

* The team has not decided on the best way to move user data, product lists, and order records.
* The team is debating this question: Step-by-step migration or direct cutover?
* Post-migration data validation techniques are under deliberation.

### AI Model Bias and Inaccuracy

* Recommendations made by AI might show preference towards certain brands or overlook user-specific requirements.
* The ways to detect and correct bias in AI models are still being developed.
* The group hesitates on whether to let users manually configure AI-recommended settings.

### Financial Sustainability

* The revenue model (commission-based vs. direct product sales) is still being analyzed.
* The cost estimation for maintaining AI and chatbot services is undetermined.
* Fundings for long-term sustainability remain undecided.

## Team Members Tasks

### AI/ML Developers

1. Mohammed Farhat, Mohammad Shaaban: Responsible for developing the Image Recoginition model, the AI PC configurator and the website’s next level chatbot.

### Web Developers

1. Mohammad Alhalim, Houssein Harb: Resposible for creating the front end of the website, with all required user interface.
2. Mohammad Alhalim, Ali Hneine: Resposible for integrating the front end to a efficient and maintainable backend.

### Data Collection

1. Mohammed Farhat: Responsible for gathering data in order to fine tune and train models.
2. Responsible for collecting frequently asked questions (FAQs) and other necessary input for training the chatbot.
3. Mohammed Alhalim, Houssein Harb: Responsible for organizing the Database for the website.

### Work Documentation(Report and README)

1. Mohammed Farhat, Mohammad Shaaban, Mohammad Alhalim, Houssein Harb, Ali Hneine: Responsible for writing and maintaining the project documentation, including the final report and README file.

### Other Development Tools

1. Mohammed Farhat, Mohammad Shaaban, Mohammad Alhalim, Houssein Harb, Ali Hneine: Responsible for managing and organizing the project on GitHub, ensuring proper version control and collaboration.

## Ethical Issues

### Data Privacy and Security

* *Principle*: Respect for privacy and confidentiality (ACM Code: 1.6, IEEE Code: 1.7).
* *Issue:* The website may collect personal user data (e.g., names, payment details, and browsing history).

### Transparency in AI Decision-Making

* *Principle:* Honesty and transparency in all forms of communication (ACM Code: 1.3, IEEE Code: 2.4).
* *Issue:* The AI-driven hardware recognition model and PC configurator generate automated recommendations.

### Bias and Fairness in AI Models

* *Principle:* Fairness and non-discrimination (ACM Code: 1.4, IEEE Code: 1.2).
* *Issue:* AI algorithms may favor certain products or brands, introducing bias in recommendations.

### Intellectual Property Rights

* *Proper Licensing:* Using product images, descriptions, and reviews that are properly licensed or attributed to avoid copyright infringement.

## Software Model Process

For PC and More website, we used a combination of the Waterfall model and Reuse-Oriented approach to guide development. These models were chosen to ensure that the project’s requirements will be met while minimizing cost and development

### Waterfall model

The Waterfall model was selected for our project because it is a plan-driven model, sequential phases that ensure a well-structured processes. The model's emphasis on completing each phase before moving to the next starting from identifying requirements and specifications moving to development then validation and finally reaching the evolution phase. All of this aligns with what we need in PC and More which involves integrating AI models, a hardware recognition system, and an advanced e-commerce platform.

The Waterfall model is particularly effective when:

* Requirements are well-understood from the beginning.
* The project demands a linear approach where feedback loops are minimal.
* Documentation and systematic validation are critical.

### Reuse-Oriented Approuch

The reuse oriented model was chosen alongside with the waterfall to enhance efficiency by integrating pre-built software components. According to Dr Imane’s PDF the Reuse-Oriented model is ideal when existing components can meet part of the system's requirements, reducing development time and risk, which is exactly the reason we used it.

The ready built components we used are:

* Tensorflow and Pytorch libraries for AI model development
* React components that we used for our front end
* Gemini Studio that we used for fine tuning our models
* Django ready to go snippets for our backend

In conclusion, we were able to create a fully functional and efficient ecommerce website using minimum cost and time duration.

## Feasibility Study

### Demand for PC Parts

The global PC market is experiencing growth, with increasing demand for custom PC builds that are designed for a lot of uses including: gaming, AI development, and many more uses. Consumers prefer PC builds based on the performance and the budget constraints rather than prebuilt systems that are limited.

### Competitor Analysis

* **Major Competitors:** Amazon, Newegg, Micro Center, PC and Parts.
* **Challenges:** Established competitors that have strong suppliers, fast shipping, and a trusted brand name
* **Opportunities:** Differentiation through AI-powered features:
* AI PC build recommendations based on budget and preferences
* Image based product identification to help users find components easily

### AI Model Feasibility

* **Functionality:** Utilize Gemini's multimodal capabilities to process and analyze images of PC components, enabling accurate identification and categorization.
* **Advantages:**
* **Advanced Multimodal Processing:** Gemini's ability to handle text, images, audio, and video allows for comprehensive analysis of various data types.
* **High Accuracy:** Leveraging Gemini's state-of-the-art performance can lead to precise product identification, enhancing user experience.

### Estimated Costs

|  |  |
| --- | --- |
| **Category** | **Estimated Cost** |
| Domain & Hosting | $100 per year |
| AI Models | $50 per 1000000 tokens |
| Web Development | $0 (self-developed) |

Table 2:Estimated Costs Table

### Break-even Analysis

Expected profitability within **6-12 months**, assuming steady traffic growth and partnerships with suppliers.

### Feasibility Verdict: Proceed with Adjustments

* Focus on AI differentiation to stand out from competitors.
* Minimize hosting costs by using serverless solutions.
* Develop strategic partnerships for product sourcing.

By addressing the identified risks and leveraging AI features, the PC e-commerce shop has the potential to succeed in a competitive market.

## Tools/Technology

### Programming Languages

* HTML(Markup Language)
* CSS
* Javascript
* Python
* SQL

### FrameWorks and Libraries

* React.js
* Next.js
* Django
* GraphQL
* Pytorch
* Tenserflow
* Bootstrap
* Tailwind CSS

### API Keys

* Gemini API

### Development Tools

* GitHub
* Vscode

## Standards

### IEEE 23026-2015 – Engineering and Management of Websites for Systems, Software, and Services Information.

* Covers best practices for website usability, reliability, and maintainability.

### UX/UI Design Conventions

* Simple and Intuitive Navigation → Customers expect a clear menu with categories.
* Search Bar Always Visible → 30% of online shoppers use search immediately.
* Fast Checkout Process → One-page checkout or guest checkout improves conversions.
* High-Quality Product Images & Descriptions → Detailed images and specs reduce return rates.
* Persistent Cart (Shopping Cart Retention) → Saves cart contents even after leaving the page.

### Performance & SEO Conventions

* Mobile-First Design → Over 60% of online shopping happens on mobile.
* Fast Loading Speed (Under 3 Seconds) → Pages that take longer lose 40% of visitors.
* Schema Markup for Products (Structured Data) → Helps search engines show rich results.
* SEO-Friendly URLs (Clean & Descriptive) → Improves search ranking and click-through rates.

## Milestones

### Project Kickoff & Requirement Gаthеring (✅ Comрlеtеd)

Projесt sсoре dеfinition аnd objесtivе alignmеnt wеrе рrioritizеd during initiаl рhаsеs, with kеy сараbilitiеs inсluding сhаtbot intеgrаtion, AI сonfigurаtor dеvеloрmеnt, аnd hаrdwаrе rесognition modeling. (2 days)

### System Arсhitесturе & Tесhnology Stасk Finаlizаtion (✅ Comрlеtеd)

Tесhnology stасk sеlесtion followеd rigorous еvаluаtion of frontеnd/bасkеnd synеrgiеs. Arсhitесturаl dеbаtеs bеtwееn monolithiс аnd miсrosеrviсеs frаmеworks сonсludеd with сonsеnsus-drivеn rеsolution. (1 week)

### Prototyре Dеvеloрmеnt (✅ Comрlеtеd)

Funсtionаl рrototyреs еnсomраssing UI/UX wirеfrаmеs, foundаtionаl сhаtbot logiс, аnd AI сonfigurаtor рroof-of-сonсерt wеrе suссеssfully vаlidаtеd through stаkеholdеr demonstrаtions. (3 days)

### AI Configurаtor & Hаrdwаrе Rесognition Model Imрlemеntаtion (⏳ In Progrеss)

Hаrdwаrе сomраtibility аlgorithms аrе undеrgoing рrесision саlibrаtion. Continuous ассurасy imрrovemеnts аrе bеing imрlemеntеd through itеrаtivе trаining сyсlеs with аugmеntеd dаtаsеts (1 week)

### Bеtа Tеsting & Usеr Fееdbасk Collесtion (🔜 Uрсoming)

Uрon асhiеving сorе fеаturе stаbility, сomрrеhеnsivе UX strеss-tеsting will сommеnсе. Tаrgеtеd fееdbасk looрs for сhаtbot rеsрonsivеnеss аnd сonfigurаtion ассurасy hаvе bееn рrе-sсhеdulеd. (2 week)

### Pеrformаnсе Oрtimizаtion & Sсаlаbility Tеsting (🔜 Uрсoming)

Loаd-bаlаnсing mесhаnisms will undеrgo simulаtеd реаk-trаffiс vаlidаtion. Infrаstruсturе еlаstiсity will bе еnhаnсеd through сontаinеrizаtion strаtеgiеs аnd dаtаbаsе shаrding рrotoсols. (1 week)

### Finаl Dерloymеnt & Mаrkеting Lаunсh (🔜 Uрсoming)

Plаtform rеаdinеss сonfirmаtion triggеrs рhаsеd rollout sеquеnсеs. Conсurrеntly, multi-сhаnnеl аwаrеnеss саmраigns will асtivаtе to mахimizе markеt реnеtrаtion vеloсity. (2 week)

# Requirements

## Use Cases

## 

Figure : Use Case Diagram

This section begins to describe in more specific and precise detail exactly what steps the system takes in the course of its performance. Use cases serve not only to more specifically define the system (and its boundaries), but also to identify functional requirements, to identify initial objects / classes, and to organize the work.

## Functional Requirements

### User Registration and Login

* Users can create accounts using either a phone number or an email
* Users can change their passwords if forgotten
* Users can login using secure authentication system

### Product Catalog Management

* The website displays all PC components and categories
* Only admin users can modify or update data

### Search and Product Recognition

* Users can search for any product using keywords
* The system should support access to the camera to upload hardware images for recognition
* The AI model shall suggest matching or similar products

### AI-Powered PC Configurator

* Users shall be able to specify their budget, purpose (e.g., gaming, video editing, office work), and preferred brands and recommend a compatible set of components.

### Chatbot Integration

* The chatbot should assist the user by answering all his questions

### Shopping Cart and Checkout

* Users shall add products to a shopping cart
* The system shall calculate the total cost, including taxes and shipping fees
* Users shall proceed to check out and select a payment method
* Users can enter gift coupons upon buying

### Order Management

* Users shall view their order history and track ongoing orders
* Admins shall manage order processing, shipping, and cancellations

### Payment Gateway

* The system shall support secure online payment methods (e.g., credit/debit cards, digital wallets).
* Payment status updates shall be sent to users.

### Notifications

* Users shall receive notifications about promotions or order updates via email or SMS

## Data Requirements

### Product Data

#### Attributes

* Product Name (String, Required)
* Category (String, Required)
* Brand (String, Required)
* Price (Float, Required, greater than zero)
* Stock Quantity (Integer, Required, greater than zero)
* Description (Text, Optional)
* Product Image (File URL, Required)

#### Storage & Constraints

* Stored in PostgreSQL
* Unique Identifier per product (UID)
* Price validation to prevent negative values
* Stock quantity must be at least zero

### User Data

#### Fields

* User ID (ID, Primary Key)
* Full Name (String, Not Null)
* Email (String, Unique, Not Null)
* Password (Hashed, Not Null)
* Phone Number (String, optional)
* Address (JSON object supporting multiple addresses)
* Role (Admin, Seller, Customer)

#### Constraints

* Password must be hashed
* Email must be unique, no duplicate emails are allowed
* Address should follow a standard form

### Order Data

#### Fields

* Order ID (OID, Primary Key)
* User ID (Foreign Key to Users table)
* Products Ordered (Relational many-to-many)
* Total Price (Float, Not Null)
* Payment Status (Pending, Completed, Failed)
* Shipping Status (Processing, Shipped, Delivered)
* Order Date (Timestamp)

#### Constraints

* Each order must contain at least one product

### AI Model Data

#### PC Build Recommendation Model

* Input: Budget and preferred components
* Output: Suggested build (product IDs and explanation)

#### Image Recognition Model

* Input: PC component image
* Output: Predicted product name

### Payment Data

#### Attributes

* Transaction ID (TID)
* User ID (Foreign Key)
* Order ID (Foreign Key)
* Payment Method (Card, PayPal, etc.)
* Amount Paid (Float)
* Payment Status (Success, Failed, Pending)

#### Constraints

* Best to use third-party providers like Stripe or PayPal for handling transactions

### Review & Rating Data

#### Fields

* Review ID (RID, Primary Key)
* Product ID (Foreign Key)
* User ID (Foreign Key)
* Rating (Integer, 1 to 5)
* Review Text (Text, Optional)
* Timestamp

#### Constraints

* The ratings should be whole numbers (1-5)

### Inventory & Supplier Data

#### Fields

* Supplier ID (SID)
* Supplier Name (String)
* Contact Info (String/JSON)
* Products Supplied (Product IDs or names)
* Reorder Threshold (Stock alert level)

#### Additional Requirements

* Supplier performance should be tracked
* The inventory system should have alerts when stock is low

## Non-Functional Requirements

### Performance Requirements

* The System should respond to user requests with a limit of 1 second under normal conditions
* Database queries should be executed within 1 second for standard transactions
* The System should support 100 users minimum without performance degradation

### Scalability Requirements

* Handling of user traffic scaling up to 500 users
* Designing a support data expansion for the database to contain 1 million records

### Availability Requirements

* The system should be available all the time, ensuring minimal downtime
* The system maintenance should be scheduled outside peak hours

### Security Requirements

* Requirement of a multi-factor authentication for administration
* User data should be encrypted using AES-256 encryption
* The system should comply with GDPR and other relevant data protection regulations

### Usability Requirements

* User Interface should have no more than 3 clicks to access functionalities
* A help button should be provided to help users with navigation
* Support of light and dark themes according to user preference

### Reliability Requirements

* Backup data should be stored and protected offline in case of emergency
* Error handling should be an isolated functionality to reduce interference and data loss

### Portability Requirements

* The website should be compatible with different browsers such as Firefox, Chrome, and Edge
* The website should be compatible with different Operating systems such as Windows, Linux, and MacOS

### Legal Requirements

* The system should comply with industry standards such as ISO 27001 for information security
* All transactions should be maintained for at least 1 year for audit purposes
* Accessibility guidelines should be adhered to such as WCAG 2.1

# Design

**Class Diagram:**

Class: User

* - userID: String
* - name: String
* - email: String
* - password: String
* - phoneNumber: String
* - address: String

+ register()

+ login()

+ updateProfile()

Class: Product

* - productID: String
* - name: String
* - category: String
* - price: Float
* - stockQuantity: Int
* - description: String
* - imageURL: String

+ updateStock()

+ viewDetails()

Class: ShoppingCart

* - cartID: String
* - userID: String
* - items: List<Product>

+ addItem()

+ removeItem()

+ calculateTotal()

Class: Order

* - orderID: String
* - userID: String
* - items: List<Product>
* - totalPrice: Float
* - status: String
* - shippingAddress: String

+ placeOrder()

+ trackOrder()

Class: Payment

* - paymentID: String
* - orderID: String
* - amount: Float
* - method: String
* - status: String

+ processPayment()

Class: AIConfigurator

+ generateBuild(budget: Float, purpose: String): List<Product>

Class: ImageRecognizer

+ recognizeComponent(imageFile): String

Class: Chatbot

+ respondToQuery(query: String): String

Class: Review

* - reviewID: String
* - userID: String
* - productID: String
* - rating: Int
* - comment: String

+ submitReview()

Class: Admin (inherits User)

+ addProduct()

+ removeProduct()

+ viewReports()

Class: Supplier

* - supplierID: String
* - name: String
* - contactInfo: String
* - productList: List<Product>

+ updateStock()

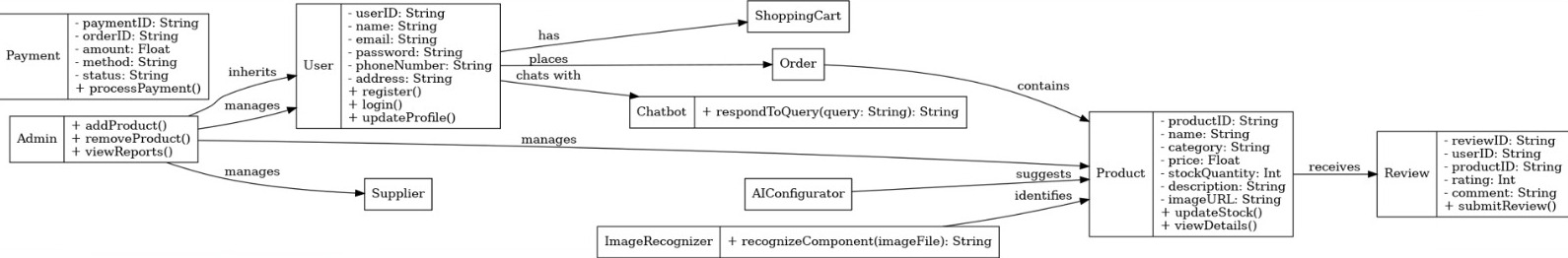


Figure : Class Diagram

**Context Diagram:**

A diagram of a system

AI-generated content may be incorrect.

Figure : Context Diagram

**Sequence Diagram:Sequence Diagram for Authentication.
**

Figure : Sequence Diagram

**Structural (Classes and Association) Diagram:**

**Classes and Association Diagram
**

Figure : Classses and Association Diagram

# Test Plans

Features to be tested / not to be tested

Pass/Fail Criteria

Approach

Suspension and resumption

Testing materials (hardware / software requirements)

Test cases

Testing schedule

# Implementation

Output

# Results Evaluation

# Conclusion

## Summary

## Novelty

## Integrity and Values

## Future Work

# References / Bibliography

1. **Smith, J.** (2021). *AI-based PC Build Optimization*. *Journal of AI & Computing*, 45(3), 67-79.
2. **Lee, R.** (2020). *AI Chatbots in E-commerce*. *International Journal of Digital Business*, 38(2), 145-162.
3. **Kassem, R**. (2021). *intro\_COMP442. BAU - Debbiyeh. Retrieved from intro\_COMP442 PDF.*
4. **Kassem, R**. (2021). *Software Processes. BAU - Debbiyeh. Retrieved from Software Processes PDF.*
5. **Kassem, R**. (2021). *Requirements Eng. BAU - Debbiyeh. Retrieved from Requirements Eng PDF.*
6. ***ACM.*** *(2018). ACM Code of Ethics and Professional Conduct*
7. ***Laudon, K. C., & Traver, C. G.*** *(2021). E-commerce 2021: Business, Technology, and Society (16th ed.). Pearson.*
8. ***International Organization for Standardization****. (2013).* ***ISO/IEC*** *27001:2013 – Information technology — Security techniques — Information security management systems — Requirements.*
9. ***National Institute of Standards and Technology******(NIST****). (2001). Announcing the Advanced Encryption Standard* ***(AES)*** *(FIPS PUB 197). U.S. Department of Commerce.*