



USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S. Computer Science

FINAL YEAR PROJECT REPORT

Batch-2021

SPECWISE

Intelligent Pc Hardware Suggestions

By

Osama Khursheed	21B-159-CS
Shaheer Muhammad Shahbaz	21B-060-CS
Shaheer Adil	21B-095-CS
Almirah Naeem	21B-138-CS

Supervised by

Muhammad Timur

Submission Performa

Name

- (1).....(Osama Khursheed)
- (2).....(Shaheer Muhammad Shahbaz)
- (3).....(Shaheer Adil)
- (4).....(Almirah Naeem)

Address

- (1).....(North Karachi, Sector 5k, Alhamra Avenue)
- (2).....(Bahria Town Karachi, Precint 11-b)
- (3).....(block 15 FB Area Karachi)
- (4).....(Block B North Nazimabad, Raees Villas.)

Email and cell

- (1).....(osamakhursheedc27@gmail.com), (03322151028)
- (2).....(shah.az0912@gmail.com), (03172887479)
- (3).....(adilshaheer09@gmail.com), (03182272607)
- (4).....(Almirahnaeem74@gmail.com), (03312688251)

SPECWISE: Intelligent Pc Hardware Suggestions

Muhammad Timur

This report is submitted as required for the Project in accordance with the rules laid down by the Usman Institute of technology as part of the requirements for the award of the degree of Bachelor **Computer Science**. I declare that the work presented in this report is my own except where due reference or acknowledgement is given to the work of others.

Signatures of students

Date

- (1).....
- (2).....
- (3).....
- (4)

Acknowledgments

Firstly, we are truly grateful to Allah Almighty. This project would not be possible without his blessings and mercy.

We extend our most heartfelt thanks to our supervisor Muhammad Timur for his invaluable help, support and guidance throughout this project.

We wish to express our thanks to our parents and friends for their encouragement and support during tough times. We stayed motivated to work on this project through their encouraging words.

Finally we would like to thank everyone who was involved in the production of this project. Your support has been highly beneficial for us.

Abstract

SpecWise is an AI-powered web service that recommends the best customized PC hardware based on certain software requirements and associated tasks. It makes use of pre-trained Large Language Models (LLMs), machine learning techniques, and web scraping from e-commerce websites. The system recognizes the issues that users can have from the prompt that the user provides and attempts to make the decision-making process simpler. These issues are more noticeable for non-technical users who might not be knowledgeable in the specifics of hardware specifications.

This project also incorporates a comparison engine that may assess the hardware choices that the LLM has suggested. This comparison is predicated on factors including cost, performance, and user-specific requirements. This enables the customer to select readily available gear that fits within their budget. This eliminates the need to sort through the several websites. Additionally, it offers a user-friendly interface that is simple enough for users of all technical backgrounds to understand.

Looking forward, Specwise seeks to boost its proficiencies through employing more sophisticated web scraping abilities to assimilate additional databases. We also aim to include features that can provide recommendations based on already existing hardware a user might have. In conclusion, this project aims to simplify the hardware recommendation process by providing the right hardware for your pc builds that balance cost and efficiency.

Keywords: PC hardware, AI, machine learning, Large Language Models, web scraping, optimization

Table of contents

Contents

Acknowledgments	3
Abstract.....	4
Table of contents.....	5
List of figures.....	9
List of Symbols	10
1 Introduction.....	13
1.1 Overcoming the challenges of pc building	13
1.2 Specwise: Hardware Recommendation system	13
1.3 Sustainable Development Goals	14
1.4 System Architecture.....	14
2 Background and Literature Review	15
2.1 Background.....	15
2.2 Similar Technologies	16
2.2.1 <i>Pcpartpicker</i>	16
2.2.2 <i>Logical increments</i>	17
2.2.3 <i>BuildMyPC</i>	18
2.2.4 <i>Userbenchmark</i>	19
2.2.5 <i>ChooseMyPC</i>	20
2.3 Summary of the related technologies	21
Key Insights	21
2.4 Machine Learning and Artificial Intelligence	22
Current State of Knowledge	22
Application of Mistral Model	22
2.5 Web scrapping for data collection	23
2.6 Framework and Technology Stack	24
2.7 Conclusion	25
3 Aim and Statement of Problem	26
3.1 Aim	26
3.2 Statement of Problem	26
3.3 Scope	26
3.4 Tackling Problems	27
3.4.1 <i>Users</i>	27
3.4.2 <i>E-commerce Platforms</i>	27
3.5 Technology Challenges	27
3.6 Changes in Aims.....	27
4 Hardware, Software analysis and requirements	29
4.1 Fact-Finding Techniques	29
4.1.1 <i>Market Research</i>	29
4.1.2 <i>Informal Discussions</i>	29
4.1.3 <i>Online Research</i>	29

4.1.4	<i>Specialized Hardware Used.....</i>	30
4.1.5	<i>Software Requirement.....</i>	30
4.2	Programming Environment	30
4.3	Software Analysis.....	31
4.3.1	<i>Django Framework (Backend)</i>	31
4.3.2	<i>React (Frontend).....</i>	32
4.3.3	<i>SQLite (Database)</i>	32
4.3.4	<i>Hugging Face API and Mistral-7B LLM.....</i>	33
4.4.5	<i>Python Libraries.....</i>	33
4.4.6	<i>React Libraries</i>	34
4.3.7	<i>HTML, CSS, and JavaScript</i>	34
4.4	Requirements in User Story Format	35
5	Software design and modeling	36
5.1	Project diagrams	36
5.1.1	<i>Class Diagram.....</i>	36
5.1.2	<i>ERD Diagram.....</i>	37
5.1.3	<i>Sequence Diagram</i>	38
5.1.4	<i>Activity Diagram.....</i>	39
5.1.5	<i>Object Diagram</i>	40
5.2	User Interface	41
5.2.1	<i>Landing Page</i>	41
5.2.2	<i>Login Page.....</i>	42
5.2.3	<i>Sign up Page.....</i>	43
5.2.3	<i>About us.....</i>	44
5.2.4	<i>Contact us.....</i>	45
5.2.5	<i>Reviews Page</i>	46
5.2.6	<i>Review History Page.....</i>	47
5.2.7	<i>Review History Panel</i>	48
5.2.8	<i>Recommendation page</i>	50
5.2.9	<i>Admin Panel.....</i>	52
5.2.10	<i>Review Panel</i>	53
5.2.11	<i>User History Panel.....</i>	54
5.2.12	<i>Products Panel</i>	55
5.2.13	<i>Recommendation Details Panel.....</i>	56
5.2.14	<i>Web scraping Panel.....</i>	57
5.2.15	<i>Web scraping Logs Panel.....</i>	58
5.2.16	<i>Web scraping Archive Panel.....</i>	59
6	Algorithm analysis and complexity	60
6.1	Web Scraping	60
6.1.1	<i>Page Traversal and Data Extraction:</i>	60
6.1.2	<i>Human-like Scrolling:</i>	61
6.1.3	<i>Data Deduplication and Change Detection:.....</i>	61
6.1.4	<i>Complexity Analysis</i>	62
6.2	LLM Prompt System	62
6.2.1	<i>Prompt classification.....</i>	62
6.2.2	<i>Recommendation Extraction.....</i>	63
6.2.3	<i>Hardware Recommendation</i>	64
6.2.4	<i>Complexity Analysis</i>	65
7	Implementation	67
7.1	Hardware Recommendation System.....	67
7.1.1	<i>LLM-Based Prompt Processing.....</i>	67
7.1.2	<i>Hardware Generation using LLM</i>	68

7.1.3	<i>Parallel Recommendation Fetching</i>	69
7.1.4	<i>Data Storage and API Integration</i>	70
7.2	Web Scraping System	70
7.2.1	<i>Scraping Strategy</i>	70
7.2.2	<i>Categories Covered</i>	71
7.2.3	<i>Key Features of the Scraper</i>	71
7.2.3	<i>Change Detection and Deduplication:</i>	72
7.2.4	<i>Automated Logging:</i>	72
7.2.5	<i>Flexible File Output:</i>	73
7.2.6	<i>Execution Workflow</i>	73
7.3	Technical Interface	73
7.3.1	<i>Fuzzy Search Hardware API Interface:</i>	73
7.3.2	<i>Sync Products interface</i>	75
7.3.3	<i>Search Hardware interface</i>	76
7.4	Architecture Diagrams	76
7.4.1	<i>Operational Flow diagram</i>	76
7.4.2	<i>Component Diagram</i>	78
7.4.3	<i>Deployment Diagram</i>	79
8	Testing	80
8.1	White Box test cases.....	80
8.2	Black Box Test cases.....	82
9	Discussion	83
10	Conclusions	86
11	Future work	88
12	Achievements	91
13	Appendices	93
13.1	Appendix A	93
13.1.1	<i>Introduction</i>	93
13.1.2	<i>Background</i>	93
13.2	Appendix B.....	94
13.2.1	<i>Tools Used</i>	94
13.2.2	<i>Similar Websites</i>	95
13.3	Appendix C.....	95
13.3.1	<i>Web scraping snippet:</i>	95
13.3.2	<i>LLM Prompt Classification Snippet</i>	97
14	References	98

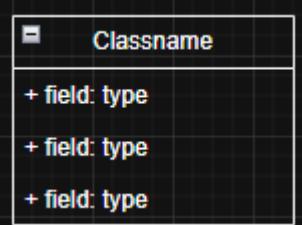
List of tables

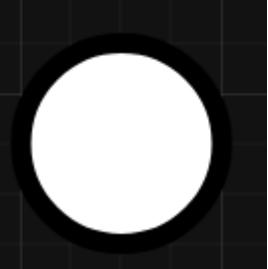
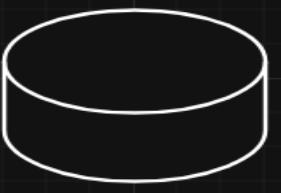
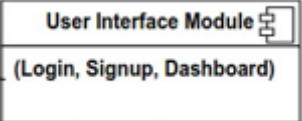
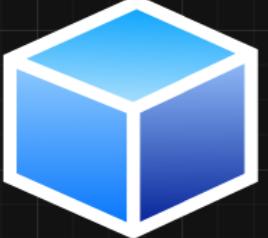
Table 1: Comparison of similar technologies.....	21
Table 2: Tools used.....	30
Table 3: Webscrape Complexity Analysis	62
Table 4: LLM Prompt Complexity Analysis.....	65
Table 5: White Box Test Cases	80
Table 6: Black Box Test Cases	82

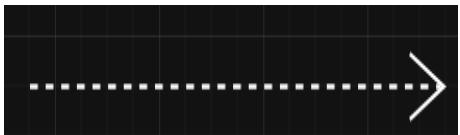
List of figures

Figure 1: PCPartPicker dashboard	16
Figure 2: Logical Increments dashboard	17
Figure 3: BuildMyPC dashboard.....	18
Figure 4: UserBenchmark Dashboard	19
Figure 5: ChooseMyPC dashboard.....	20
Figure 6: Class Diagram.....	36
Figure 7: Entity Relationship Diagram	37
Figure 8: Sequence Diagram	38
Figure 9: Activity Diagram	39
Figure 10: Object Diagram.....	40
Figure 11: Specwise Landing page	41
Figure 12: Specwise Login page	42
Figure 13: Specwise Signup page	43
Figure 14: Specwise About Us Page	44
Figure 15: Specwise Contact Page	45
Figure 16: Specwise Review page.....	46
Figure 17: Specwise Review History Page	47
Figure 18: Specwise Recommendation History Page	48
Figure 19: Specwise Recommendation Page Details	48
Figure 20: Specwise Recommendation Page Details	49
Figure 21: Specwise Recommendation Page	50
Figure 22: Specwise Admin Panel	51
Figure 23: Specwise Review Admin Panel	52
Figure 24: Specwise Question History Panel	53
Figure 25: Specwise Products Panel	54
Figure 26: Specwise Recommendation Details Panel	55
Figure 27: Specwise Web scraping Gui	56
Figure 28: Specwise Web scraping Logs Panel	57
Figure 29: Specwise Web scraping Archives Panel.	58
Figure 30: LLM recommendation operational diagram	76
Figure 31: web scraping Operational Diagram	77
Figure 32: Component Diagram.....	78
Figure 33: Deployment Diagram.....	79

List of Symbols

No.	Name	Symbol	Description
1	Class		Represents a class with name, attributes, and methods
2	Association		A relationship between two classes
3	One to one (1:1)		One entity in A relates to only one entity B
4	One to many (1:N)		One entity in A relates to multiple entities in B
5	Many to many (N:N)		Many entities in A relate to many in B.
6	Actor		Represents external user or system.
7	Object		Represents an object and its time axis.
8	Return Arrow		Return of control after message.

9	Start Node			Entry point of the activity.
10	Decision			Represents a condition-based branch.
11	Activity			Represents a task or step.
12	Database			Represents a structured data store.
13	Input/Output			Input from or output to external source.
14	Component			Represents a software component/module.
15	Process			Represents a task or action.
16	Device			Represents a hardware or execution environment.

17	Dependency		Shows that a component depends on another.
----	------------	---	--

1 Introduction

1.1 *Overcoming the challenges of pc building*

The SpecWise project aims to address a challenge faced by the majority of PC building users: determining which hardware components are suitable for executing specific software applications. The majority of solutions on the market suggest characteristics that are general to consumers' needs, which results in disappointing outcomes. This is a frustrating condition for many users, especially those who lack technological ability. This could lead to subpar build performance or, in some cases, unnecessary expenses.

1.2 *Specwise: Hardware Recommendation system*

What distinguishes SpecWise from other PC building websites is the use of artificial intelligence, specifically pre-trained Large Language Models, to select hardware components. The proposed system tackles this problem by allowing the user to specify types of software (Adobe Photoshop, AutoCAD, etc.) and tasks (e.g., video editing, high-resolution graphics work) as a prompt. After processing the prompt and recommending hardware, the LLM compares the results and uses web scraping to extract up-to-date data from several e-commerce websites, including prices and reviews, before producing the final output. Based on a study of the requirements and skills of each user, these guidelines offer a set of hardware recommendations.

1.3 *Sustainable Development Goals*

This project doesn't aim only to improve the users' PC-building practice but also contributes to multiple Sustainable Development Goals (SDGs).

- SDG Goal 9: Industry, Innovation, and Infrastructure is supported through SpecWise's approach of integrating AI and machine learning to provide optimized hardware recommendations, promoting technological advancement in the PC-building ecosystem.
- SDG Goal 10: Reduced Inequality, which guarantees that people with different levels of technical expertise can still make wise choices.
- SDG Goal 12: Responsible Consumption and Production is also fulfilled by encouraging users to purchase only the hardware they need, reducing overconsumption and unnecessary waste that can be made with poor decisions.
- SDG Goal 8: Decent Work and Economic Growth by helping professionals choose optimized hardware that enhances their productivity, contributing to economic growth.

1.4 *System Architecture*

A pre-trained LLM and an API for web scraping are part of the system design. The LLM compares factors including CPU and GPU performance, pricing, and energy economy of various systems to provide recommendations. These elements guarantee that the suggested hardware satisfies user requirements while maximizing cost and performance.

2 Background and Literature Review

2.1 *Background*

Many users prefer building their own PCs over buying prebuilt PCs to ensure customization and quality [1]. Building a PC can be a tedious task, especially for those who lack basic technical knowledge. Selecting the right components can be a very big challenge for them, due to the rapid evolution of the PC hardware industry, which frequently sees the release of new and updated components. Since the early years of PC development, the industry has experienced exponential growth, with purchases increasing by two orders of magnitude in just six years [2]. Research indicates that the average consumer struggles to keep pace with these advancements; technological complexity is identified as the primary source of confusion, leading to overchoice confusion [3]. Many individuals might struggle to keep up with these advancements, highlighting the need for PC building tools.

This chapter will explore similar technologies and their features as well as some of the tools that are employed in this project.

2.2 Similar Technologies

2.2.1 Pcpaprtpicker

One of the most popular website used for pc builds. It has a very large database of components and provides the basic tools needed to virtualize pc builds. [4]

Features:

- Simplified building tool
- Price tracking
- Compatibility checks
- Benchmarks
- Completed builds

The screenshot shows the PCPartPicker dashboard interface. At the top, there is a header with a logo, a search bar containing the URL <https://pcpartpicker.com/list/sF8TwP>, and various navigation and settings icons. Below the header, a green banner displays a compatibility check message: "Compatibility: No issues or incompatibilities found." and an estimated wattage: "Estimated Wattage: 0W". The main area is a table where users can select components. The columns are labeled: Component, Selection, Base, Promo, Shipping, Tax, Price, and Where. The rows represent different components: CPU, CPU Cooler, Motherboard, Memory, Storage, Video Card, Case, Power Supply, Operating System, Monitor, Expansion Cards / Networking, and Peripherals. Each row has a "Choose" button. The "Expansion Cards / Networking" row lists "Sound Cards, Wired Network Adapters, Wireless Network Adapters". The "Peripherals" row lists "Headphones, Keyboards, Mice, Speakers, Webcams".

Figure 1: PCPartPicker dashboard

2.2.2 Logical increments

Provides straightforward and tiered based recommendations. It has extensive guides on every components for more technical users and provides a forum where users can ask questions about their builds. [5]

Features:

- Extensive information and guides on components
- Features blogs of new components in the market
- Visual build guides
- User reviews

	Graphics	CPU	CPU Cooler	Motherboard	RAM	HDD	SSD	Power Supply	Case	Total
Destitute	UHD 610 \$0	G5905 \$40	Stock \$0	ASRock B560M-HDV \$104	4GB DDR4 \$14	None \$0	256GB \$26	Tt 500W (W) \$40	Focus G \$55	\$279
		G5900 \$48	212 Black \$30	ASUS H510M-E \$136	8GB DDR4 \$21	2TB \$52	512GB \$48	SeaSonic 500W (B) \$65	MX330X \$64	
	▼	▼	▼	expand to see details				▼	▼	▼
Poor	Vega 8 \$0	R3 3200G \$80	Stock \$0	ASUS B450M-A \$70	4GB DDR4 \$14	None \$0	256GB \$26	Tt 500W (W) \$40	Focus G \$55	\$285
			AK400 \$35	ASRock B450M-HDV \$73	8GB DDR4 \$21	2TB \$52	512GB \$48	SeaSonic 500W (B) \$65	MX330X \$64	
	▼	▼	▼	expand to see details				▼	▼	▼
Minimum	Vega 7 \$0	R5 5600G \$137	Stock \$0	ASUS B450M-A \$70	8GB DDR4 \$21	2TB \$65	512GB \$48	Tt 500W (W) \$40	Focus G \$55	\$436
			212 Black \$30	ASRock B450M-HDV \$73			1TB \$70	SeaSonic 500W (B) \$65	MX330X \$64	
	▼	▼	▼	expand to see details				▼	▼	▼
Entry	▲ RX 550 \$83	i3 12100F \$85	Stock \$0	ASUS H610M-A \$110	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	Focus G \$55	\$532
			Pure Rock 2 \$38				1TB \$70	SeaSonic 550W (B) \$70	Archon 2 \$60	
	▼	▼	▼	expand to see details				▼	▼	▼
Modest	RX 6400 \$120	i3 12100F \$85	Stock \$0	ASUS H610M-A \$110	8GB DDR4 \$21	2TB \$65	512GB \$48	SeaSonic 500W (B) \$65	Focus G \$55	\$569
	RX 6500XT \$140		Pure Rock 2 \$38		16GB DDR4 \$31		1TB \$70	SeaSonic 550W (B) \$70	Archon 2 \$60	
	▼	▼	▼	expand to see details				▼	▼	▼

Figure 2: Logical Increments dashboard

2.2.3 BuildMyPC

BuildMyPC is very similar to PCPartPicker in terms of features. It has an extensive list of components as well as pre built PCs and Laptops [6].

Features:

- Compatibility checks
- Cost optimization
- Build Guides

The screenshot shows the BuildMyPC dashboard. At the top, there is a header with the URL 'https://buildmypc.net/rigs/PjFBLz', social sharing icons, and a 'Select Country' dropdown set to 'United States'. Below the header, a green banner displays a message: 'Compatibility: No issues or incompatibilities found.' The main area is a table with the following columns: Component, Product, Title, Price, Product Link, and Remove. The table lists the following components with their respective '+ ADD Component' buttons:

Component	Product	Title	Price	Product Link	Remove
Processor	+ ADD Component				
Motherboard	+ ADD Component				
Cabinet	+ ADD Component				
RAM	+ ADD Component				
Storage	+ ADD Component				
Graphics Card	+ ADD Component				
Power Supply	+ ADD Component				
Cabinet Cooler	+ ADD Component				
CPU Cooler	+ ADD Component				
Monitor	+ ADD Component				

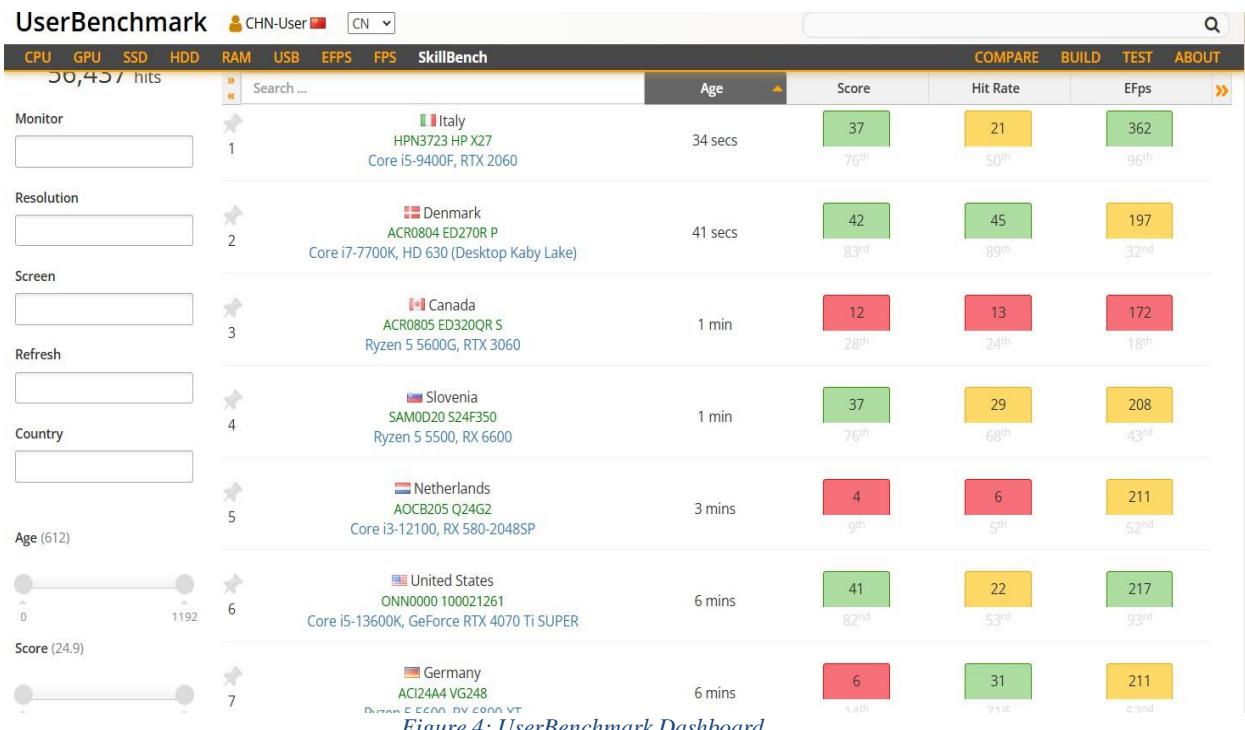
Figure 3: BuildMyPC dashboard

2.2.4 Userbenchmark

Allows for better comparisons by allowing users to test their hardware performance against preexisting builds. Although it does not provide personalized builds, it is a great way to check your build functionalities against other components on the market [7].

Features:

- Large component database
- Live benchmark tracking
- Community Insights
- Hardware recommendations



2.2.5 ChooseMyPC

It has a very simple user interface and provides a more customized way to build PCs based on the preference and the budget of the user [8]

Features:

- Budget consideration
- Simple interface
- User reviews
- Building Guides

The screenshot shows the ChooseMyPC.net dashboard. At the top, there's a navigation bar with links for Home, Build Generator, Guides and Resources, and Feedback. Below the navigation, a large input field asks "How much do you want to spend?" with a placeholder "This dictates the performance of the PC you will get, of course". A note below it says "It's just as important to avoid spending too much for your needs, as well as worrying about not having enough! You can find real-world gaming benchmarks for the graphics cards included in the builds to see approximate performance for the build". A slider is set to \$1000. Underneath, a section asks "Would you like to overclock your CPU?" with a note: "Overclocking involves the CPU at a faster clock speed than it was originally designed for. This creates more heat and wears the CPU more but yields more CPU performance". It includes a note: "Don't feel like this is essential, as you can sometimes even get a better rig for the money by not bothering and spending the money on other parts instead. You should only do it if you are confident and you are enthusiastic about overclocking". There are three buttons: "I want to overclock", "Maybe in future", and "I'd rather not". Below this is a section titled "Additional Options" with notes about customizing extras like Windows and optical drives. A "Go to Build" button is at the bottom.

Figure 5: ChooseMyPC dashboard

2.3 Summary of the related technologies

Feature / Tool	PCPartPicker	Logical Increments	Build MyPC	UserBenchmark	ChooseMyPC	SpecWise
Large Component Database	✓	✗	✓	✗	✗	✗
Compatibility Checks	✓	✗	✓	✗	✓	✓
Price Comparison	✓	✗	✓	✗	✗	✓
Tier-Based Recommendations	✗	✓	✗	✗	✗	✗
AI-Based Recommendations	✗	✗	✗	✗	✗	✓
Software-Specific Guidance	✗	✗	✗	✗	✗	✓
Cost Optimization	✓	✓	✓	✓	✓	✓
Real-Time Data Scraping	✗	✗	✗	✗	✗	✓
Performance Benchmarking	✗	✗	✗	✓	✗	✗
User-Friendly Interface	✓	✓	✓	✓	✓	✓
User Reviews/Community Insights	✓	✓	✗	✓	✓	✓

Table 1: Comparison of similar technologies

Key Insights

- PCPartPicker focuses on component compatibility and price comparisons.
- Logical Increments offers tiered recommendations but lacks personalized recommendations.
- BuildMyPC focuses on compatibility and cost-effectiveness similar to PCPartPicker.
- UserBenchmark provides benchmarking capabilities but does provide personalized hardware recommendations.
- ChooseMyPC provides a very simple user interface and offers recommendations based on budget and preferences.

Specwise aims to set itself apart by providing AI-driven recommendations that are tailored to software needs and web scrapping data in real-time.

2.4 *Machine Learning and Artificial Intelligence*

Current State of Knowledge

Machine learning is being applied into almost every smart system that handles large amounts of data. In context of hardware recommendations, Machine learning is applied in predicting hardware recommendations because of its high accuracy. ML models have been used to predict the area of hardware designs based on abstract specifications, reducing the time and complexity involved in hardware design [9]. Furthermore, ML techniques can assist in dimensioning hardware resources for AI applications while respecting user-defined constraints, ensuring that the hardware is optimized for both performance and efficiency [10].

Pre-trained language models (PLMs) and large language models (LLMs) have shown significant potential in various natural language processing tasks. These models can be effectively employed to break down and understand user prompts or commands [11]. That is why we are going to use a LLM to break down the user's specification to make easier to understand and make recommendations.

Application of Mistral Model

We plan to use Mistral 7B for this project which is available through Hugging face. It is an excellent model for this project because it produces hardware recommendations by breaking down user specifications efficiently and accurately [12].

2.5 *Web scrapping for data collection*

Web scraping is an automated technique used to extract data from websites, converting unstructured web content into structured databases [13]. We included web scrapping in this project so that we can have more freedom on which type of data we use. Also this step was taken to ensure that the prices of the components is up to date.

Different types of web scraping, including content scraping, price scraping, and database scraping, are essential for retrieving the relevant information [14]. Many types of scrappers are available to use based on specific needs like Selenium for automation of dynamic web pages [15] or Beautiful Soup (bs4), a python library that extracts data from HTML and XML documents [16]. For more complex web pages, scrapy can be used as it is suited for extracting data from more complex web pages [17] and octoparse for its no code approach [18].

Some use cases of the more prominent web scrappers include:

- **BeautifulSoup:** used for websites with static content. It allows us to quickly traverse the DOM tree and extract necessary tags.
- **Scrapy:** used to create scalable spiders with built-in support for crawling, exporting, and throttling. It is very helpful in dealing with large datasets.
- **Selenium:** used for sites that rely heavily on JavaScript and required dynamic content rendering. It is helpful for simulating user interactions like scrolling or button clicking to bypass captchas.

For SpecWise we opted to use Selenium as it allowed us to automate real browser actions such as page scrolling, dynamic element loading, and waiting for asynchronous JavaScript to finish rendering which was not easily achievable with the other available parsers.

Some challenges one can face during the web scrapping process is the frequent layout change which can cause the HTML tags of the website to change, secondly there are different formats of content for each site which might not be consistent.

The most important challenge that we have to tackle while dealing with these websites is their anti bot measures that limit the use of web scrappers and bots as the use of these actually increase the traffic of the websites. That is why many sites such as b and h photo and wallmart had anti throttling and ip banning which blocks the access to these sites completely. This is why it is very important to read the robots.txt file provided by all e commerce platforms for their policy on bot usage.

Thus, while web scraping is great alternative for data collection, it is not a one-time solution. Ongoing monitoring, ethical considerations, fallback plans, and potential API integrations will be important to ensure the continued success of platforms like SpecWise.

2.6 *Framework and Technology Stack*

A system as complex as this project would require a framework to be scalable, solid and have capabilities to handle user interactions, handle data processing and third party integrations. And after comparing common frameworks, Django proved to be the most suited for this project.

Django is a python frameworks that allows the creation of websites. Its clean, easy to setup and is widely used these days for many websites due to its community support. Its is renowned for its "batteries included" philosophy, offering a comprehensive standard library and extensive built-in features [20]. Its key strengths include its object-relational mapping (ORM) for simplified database interactions, built-in authentication and authorization systems, and emphasis on security and performance [21]. The framework adheres to the "Don't Repeat Yourself" (DRY) principle, streamlining the development of complex web applications[21]. Django's versatility, scalability, and extensive documentation have contributed to its popularity among developers and its use in numerous successful applications [22].

Furthermore, its seamless integration with SQL proved to be essential for storing data in this project. Django's Object-Relational Mapping (ORM) system offers powerful capabilities for database management and migration. It enables abstraction of data from existing databases, facilitating migration between different database management systems like MySQL and PostgreSQL [23]. Django's integration with SQL can be further enhanced for security purposes. [24]

2.7 *Conclusion*

By using LLMs and Web scraping, SpecWise seeks to close the gaps left by comparable websites and give users a more tailored experience. SpecWise strives to stay current with components in the market by utilizing web scrapers like BeautifulSoup and Scrapy and technologies like Mistral 7b to better comprehend user prompts.

In addition to offering a seamless PC building experience, Specwise also makes its platform user-friendly online so that consumers can access it with ease.

3 Aim and Statement of Problem

3.1 Aim

The aim of this project is to develop an intelligent web-based hardware recommendation system. This system will provide users with hardware suggestions based on their specific software requirements and needs. By utilizing data scraped from e-commerce websites, pre-trained Large Language Models (LLMs), and machine learning algorithms, our system ensures that users receive cost-effective and performance-optimized hardware recommendations that suits their needs.

3.2 Statement of Problem

In today's modern and technologic era, selecting the right hardware for specific software applications is a very daunting task, especially for those who are not informed enough about the recent technologies. With new hardware being released into the market almost every day, many existing hardware get overshadowed by other newer and better hardware. That is why it is almost impossible for an average person to keep up with these advancements.

3.3 Scope

The scope of the SpecWise project is to provide personalized hardware suggestions based on software requirements of the user, utilizing real-time scraped data, pre-trained LLMs and machine learning algorithms . The Users input their software requirements on the website, and the system generates recommendations, along with performance and cost comparisons. SpecWise also aims to continually update its recommendations by integrating the latest market data and user feedback.

3.4 *Tackling Problems*

3.4.1 **Users**

Users that don't have technical backgrounds often find themselves stuck and confused when selecting the right hardware. Specwise simplifies this process by generating appropriate recommendations that are tailored according to user input such as video games and graphic design. By bridging the knowledge gap, SpecWise empowers users to confidently choose components that match their unique requirements.

3.4.2 **E-commerce Platforms**

Scraping data from ecommerce platform on a weekly basis can ensure that the provided hardware and their prices are up-to-date. This prevents the need of constantly monitoring the price changes manually.

3.5 *Technology Challenges*

Real-time data scraping is challenges due to changes in e-commerce site layouts and legal issues. Most sites have strict rules for data scrapers, most commonly they don't allow more than one request per second as this generates traffic for the website. Additionally, the Large Language Model used may struggle with comparing performances of the more niche hardware. Maintaining data integrity and accuracy is a constant challenge due to the dynamic nature of online marketplaces.

3.6 *Changes in Aims*

If any changes in the project's aims arise due to technical difficulties or any sort of time constraints, such as limited access to e-commerce websites or irregularities in the comparison system, these changes will be documented as they occur.

Currently, the project focuses on delivering a web-based hardware recommendation system by using data scrapped from different e-commerce platforms. In response to unforeseen challenges, the scope may be adjusted to prioritize core functionalities and ensure a reliable user experience. Continuous evaluation and stakeholder feedback will guide any necessary modifications to the project aims.

4 Hardware, Software analysis and requirements

4.1 Fact-Finding Techniques

In this project, we employed a variety of fact-finding techniques to ensure a comprehensive understanding of user needs, market trends, and technical requirements. This multi-faceted approach allowed us to capture both the big picture and the finer details of the hardware recommendation landscape.

4.1.1 Market Research

By examining popular hardware suggestion websites such as PCPartPicker, Logical Increments, and BuildMyPC, we conducted thorough market research. This investigation helped us understand the strengths and limitations of existing platforms, their customization options, and how SpecWise can offer unique value by integrating AI-driven personalized recommendations.

4.1.2 Informal Discussions

We engaged in informal discussions with colleagues, friends, and relatives who have experience building custom PCs. These conversations provided valuable insights into the challenges users face when selecting hardware tailored to specific software needs, highlighting common pain points and expectations.

4.1.3 Online Research

To further deepen our understanding, we explored online forums, tech communities, and survey results related to PC hardware selection. This research revealed prevalent issues and user frustrations, reinforcing the need for a solution like SpecWise that simplifies and personalizes the hardware recommendation process.

4.1.4 Specialized Hardware Used

Since this project is web-based, no hardware will be required in its development. AWS and other cloud-based resources are used to manage the computing labor needed for web scraping and processing massive data volumes.

4.1.5 Software Requirement

The systems and applications required to function well are outlined in the project's software requirements. It may also include information on any external libraries, tools, or APIs that the project intends to use.

4.2 *Programming Environment*

For our project to run well and handle both frontend and backend operations, certain software tools, libraries, and platforms are needed. The programming environment, external APIs, libraries, databases, and frameworks that will be used are described in this section

Category	Tools/Technologies
Web Browsers	Google Chrome, Firefox, Microsoft Edge
Operating Systems	Windows, Linux
Frameworks	Django (Python), React (JavaScript)
Databases	SQLite

Pretrained LLM	Mistral-7B via Hugging Face
APIs	Hugging Face Transformers API for LLM
Libraries	Pandas, Scikit-learn, Fuzzy Search
Programming Languages	Python, JavaScript, HTML, CSS
IDEs	Visual Studio Code
Web scrappers	Selenium

Table 2: Tools used in the project

4.3 Software Analysis

The software environment that was selected was created to satisfy the particular requirements of the SpecWise project. The system communicates with users using a web-based interface, processes real-time data from many sources, and makes suggestions using a potent LLM. The function of each component and the rationale for its selection are explained in the section that follows.

4.3.1 Django Framework (Backend)

Django is a web framework built on Python that offers SpecWise a reliable backend. It provides capabilities for managing databases, user authentication, and form submission, all of which are essential for this project, and it facilitates quick development and clean design. User inputs, hardware suggestions, and scraped data are all stored in an intuitive relational database structure thanks to Django's default usage of SQLite.

Rationale for Django:

- Built-in support for REST APIs, essential for connecting the backend with the React frontend.
- Fast development cycle, reducing project complexity while maintaining a high level of security.
- Django's ORM (Object Relational Mapper) simplifies database management and interactions with SQLite.

4.3.2 React (Frontend)

SpecWise's frontend is built with ReactJS, which offers a responsive and dynamic user interface. Users can input their software requirements and view real-time hardware suggestions with React's smooth user interface.

Rationale for React:

- Enables a Single Page Application (SPA), providing smooth navigation and updates without reloading the page.
- Highly modular, allowing the development team to create reusable UI components.
- React integrates well with Axios for making API requests to the backend.

4.3.3 SQLite (Database)

SpecWise's database system is SQLite, which is Django's default. Despite not being built for heavy traffic, SQLite is perfect for early development and makes local testing and deployment simple. Logs, hardware recommendations, and user input are all stored in the database.

Rationale for SQLite:

- Lightweight, requiring minimal configuration, making it ideal for a web application like SpecWise.
- Integrated directly into Django, which simplifies database management.
- Scalable enough for the project's current scope, with options to switch to a more robust system (e.g., PostgreSQL) if needed.

4.3.4 Hugging Face API and Mistral-7B LLM

SpecWise makes use of Mistral-7B, an LLM that has already been trained and can be accessible using the Hugging Face API. This model generates a preliminary hardware suggestion after processing user input pertaining to software needs. Hugging Face's transformer models enable SpecWise to decipher intricate user inquiries and offer accurate recommendations according to the jobs and software input.

Rationale for Hugging Face and Mistral-7B:

- Pretrained models simplify the integration of advanced NLP without requiring extensive training.
- Hugging Face offers easy integration with Django via its API, allowing for real-time recommendations.
- Mistral-7B provides highly accurate results, which are crucial for ensuring that users receive appropriate hardware configurations.

4.4.5 Python Libraries

Several Python libraries form the backbone of the backend processing for SpecWise. These libraries handle data manipulation, machine learning, and web scraping tasks.

- **Pandas**: Used for data analysis and manipulation, especially when processing web-scraped data and preparing hardware recommendations.
- **Scikit-learn**: Supports machine learning tasks, such as refining the hardware recommendation model based on historical user data.
- **BeautifulSoup and Selenium**: Essential for web scraping, these libraries gather hardware specifications, prices, and availability from e-commerce platforms.

Rationale for Python Libraries:

- Python provides a powerful ecosystem for data processing, machine learning, and web scraping.
- **Pandas** and **Scikit-learn** allow for efficient data manipulation and model building.
- **Selenium** automates web scraping tasks, ensuring that SpecWise has up-to-date hardware information.

4.4.6 React Libraries

SpecWise uses React-based libraries to facilitate user interaction and produce a seamless, intuitive user experience.

- **Axios:** A promise-based HTTP client for making requests to the Django backend.
- **React-Router-DOM:** Enables dynamic routing, allowing users to navigate between different pages without reloading the app.
- **React Bootstrap:** For building responsive layouts and components that adapt to different screen sizes.

Rationale for React Libraries:

- **Axios** simplifies backend communication, essential for real-time interactions with the API.
- **React Bootstrap** ensures that the application is responsive and accessible across devices, enhancing usability.

4.3.7 HTML, CSS, and JavaScript

The SpecWise application must be structured and styled using HTML and CSS. JavaScript enables client-side, dynamic functionality, resulting in a smooth user experience.

Rationale for HTML, CSS, and JavaScript:

- HTML and CSS ensure a consistent user interface that is easy to navigate.
- JavaScript allows for responsive interactivity, such as real-time data display and user input validation.

4.4 Requirements in User Story Format

- **As a new user, I want** to input the software I plan to use **so that** I can receive hardware suggestions suited to my needs.
- **As a registered user, I want** to maintain my history **so that** I can make informed decisions from my past.
- **As a developer, I want** to continuously scrape hardware data from e-commerce websites **so that** the recommendations remain current.
- **As a system administrator, I want** to ensure the LLM processes user queries accurately **so that** I can provide personalized recommendations.

5 Software design and modeling

This chapter comprises of the modeling of the system using various UML diagrams.

5.1 Project diagrams

5.1.1 Class Diagram

Fig 6 shows the class diagram with user, authentication, frontend interface, webscraped data, reports and hardware recommendation system. It also displays how the classes are related to each other.

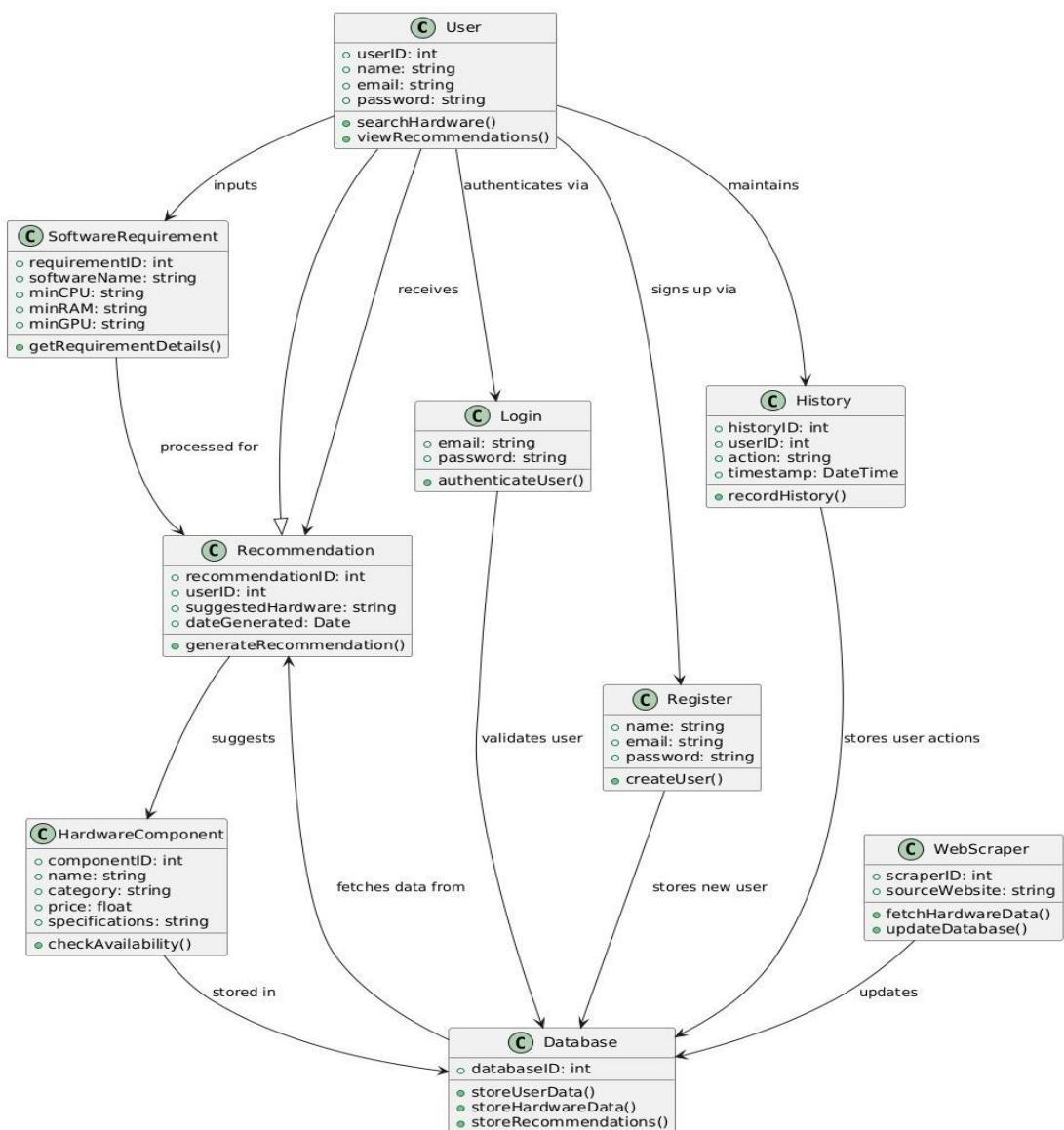


Figure 6: Class Diagram

5.1.2 ERD Diagram

ERD diagrams are mostly used to build relational databases. Fig 7 shows the sqlite database structure that the project will have.

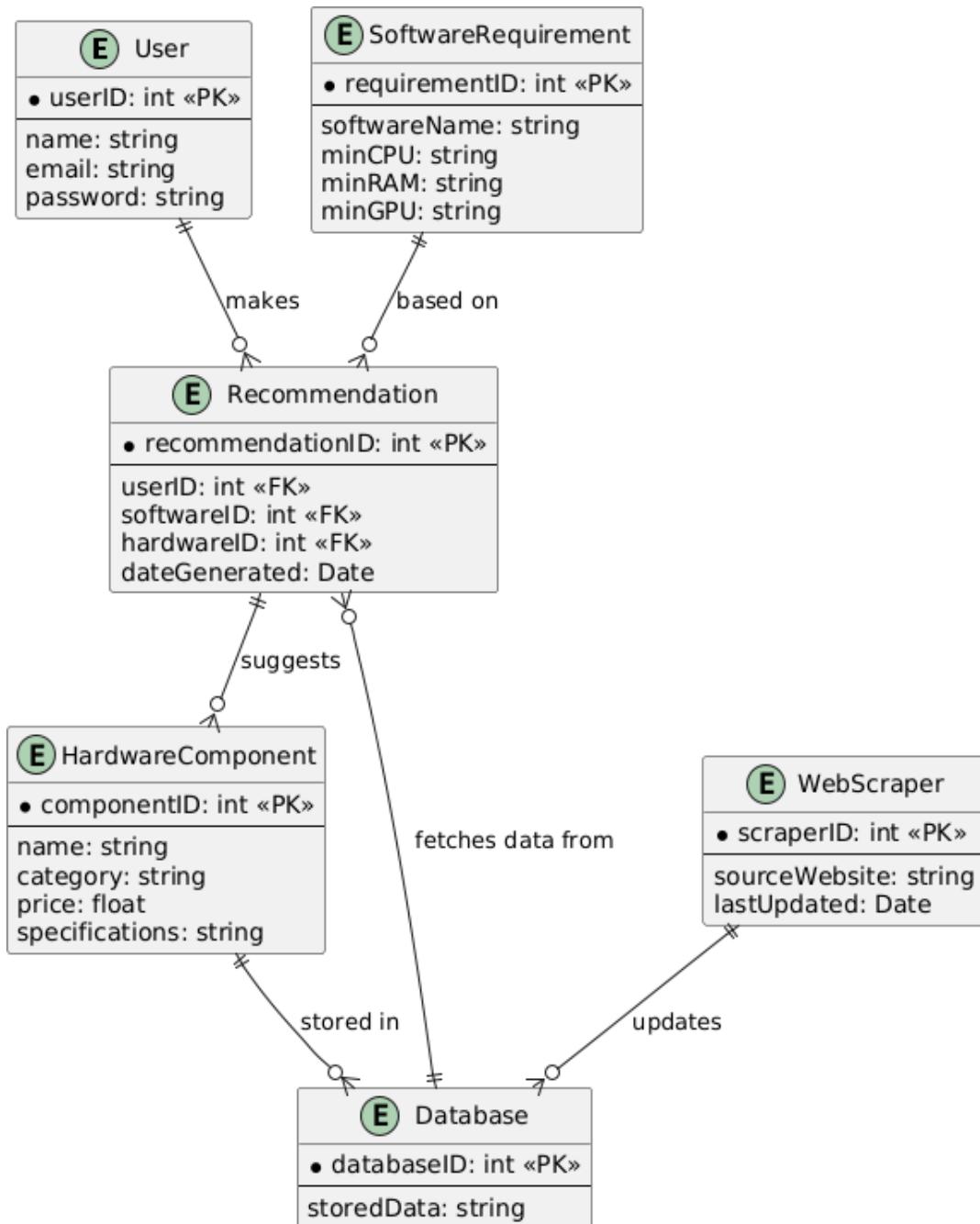


Figure 7: Entity Relationship Diagram

5.1.3 Sequence Diagram

Sequence Diagrams show how objects and classes interact with each other. Fig 8 shows how the user initiates a request and how it moves through the system.

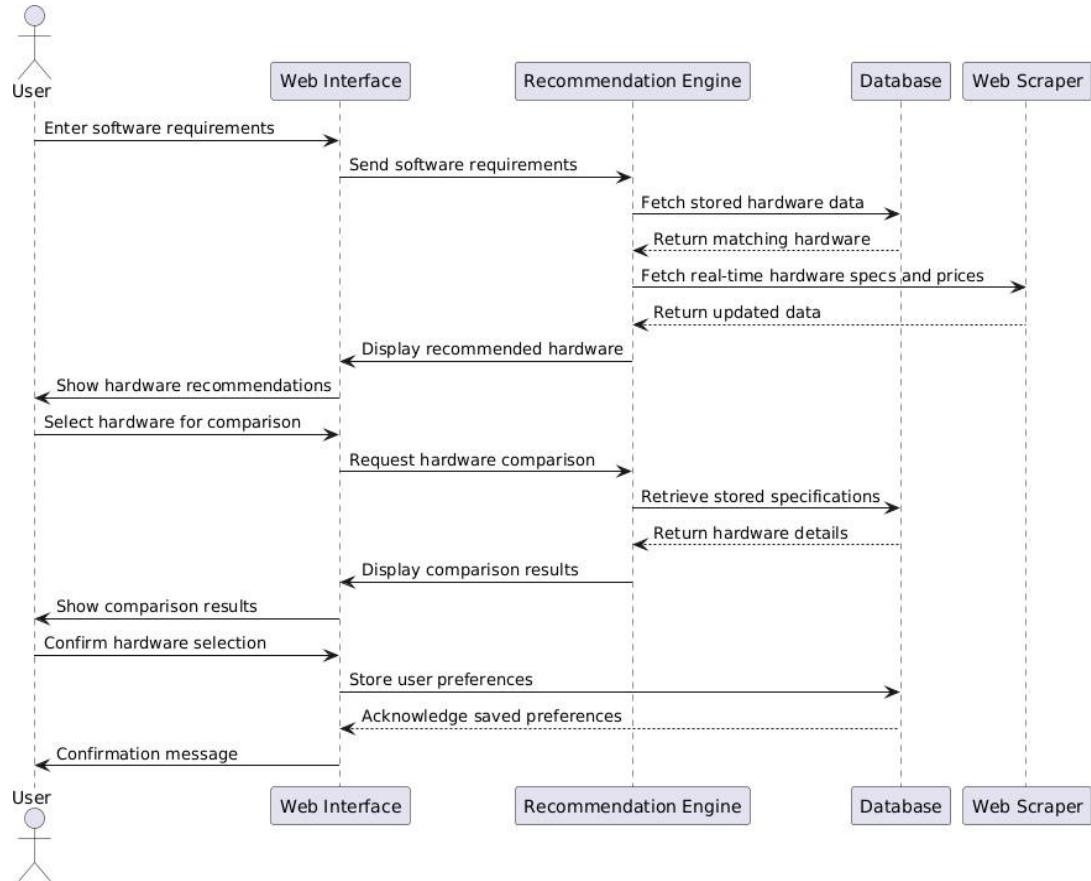


Figure 8: Sequence Diagram

5.1.4 Activity Diagram

Activity diagrams are used to show the flow of data and control in a system. Figure 9 denotes the flow of a user request through the system

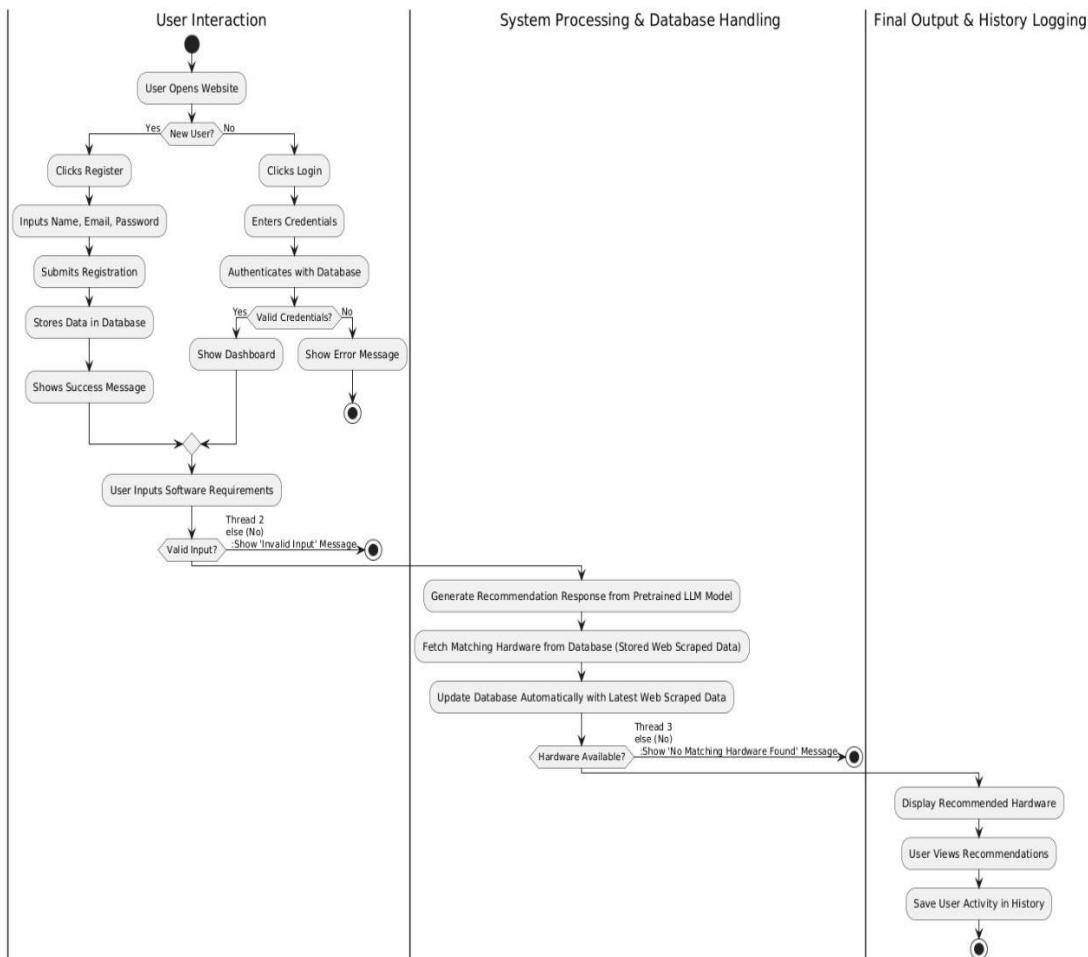


Figure 9: Activity Diagram

5.1.5 Object Diagram

An Object diagram shows the instances of classes and their relationship at a particular point in time. Fig 10 shows the object diagram of this project.

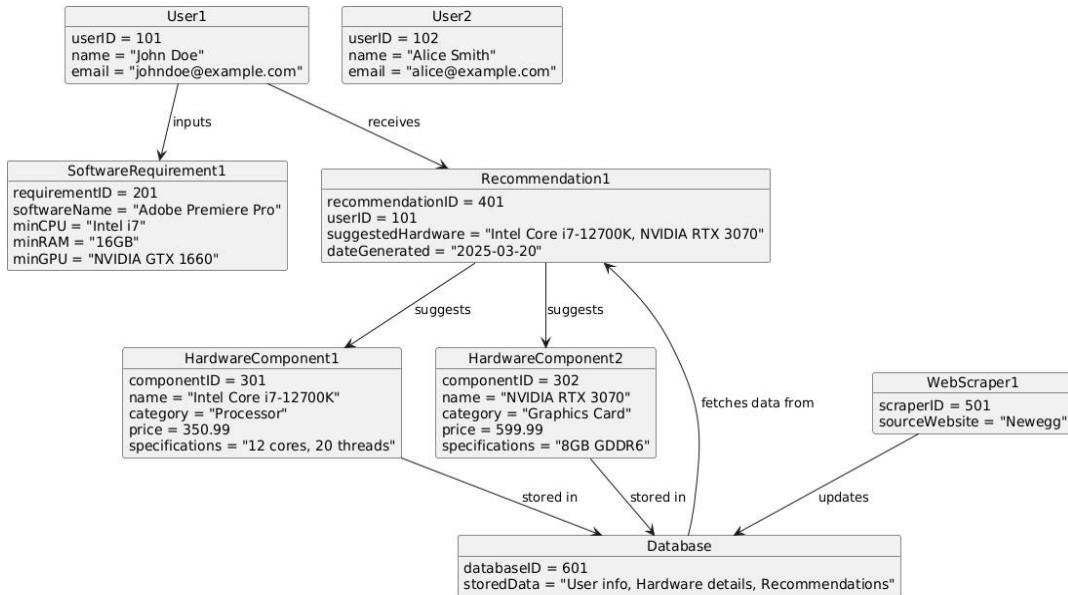


Figure 10: Object Diagram

5.2 User Interface

5.2.1 Landing Page

The screenshot displays the Specwise landing page. At the top, there is a navigation bar with links for Home, About Us, Contact Us, Review Portal, Review History, History, Welcome, shahbaz, and Logout. The main header is "Find Your Perfect PC Hardware". Below it, a sub-header says "Get tailored PC hardware recommendations based on your specific tasks and software needs." A prominent blue button labeled "Get Your Recommendations" is present. To the right of the text, there is a decorative graphic featuring a smartphone displaying the Specwise logo, surrounded by various computer hardware components like a hard drive, RAM, and a wrench.

How SpecWise Works

Get personalized PC hardware recommendations in 3 easy steps.

How SpecWise Works

Get personalized PC hardware recommendations in 3 easy steps.

Input Your Needs
Describe the software and tasks you want to perform.

AI-Powered Analysis
Our AI analyzes your input and determines the best hardware.

Receive Tailored Recommendations
Get the best options for your PC hardware.

Why Choose SpecWise?

The smarter way to choose your PC hardware.

Real-time Data Scraping
Stay updated with the latest hardware options.

AI-Driven Suggestions
Recommendations tailored to your needs.

Customizable Recommendations
Choose what suits you best.

Detailed Comparison Reports
Make informed decisions with detailed reports.

Ready to Build Your Perfect PC?

Get started with tailored hardware recommendations that match your needs.

Get Started

Frequently Asked Questions

How does SpecWise generate hardware recommendations?
What e-commerce platforms do you scrape data from?
What types of software can I get recommendations for?

Figure 11: Specwise Landing page

The Landing Page greets you with a welcoming design and a clear introduction to SpecWise. It quickly shows you what the platform does and invites you to sign up or try the website as a guest, making it easy to get started.

5.2.2 Login Page

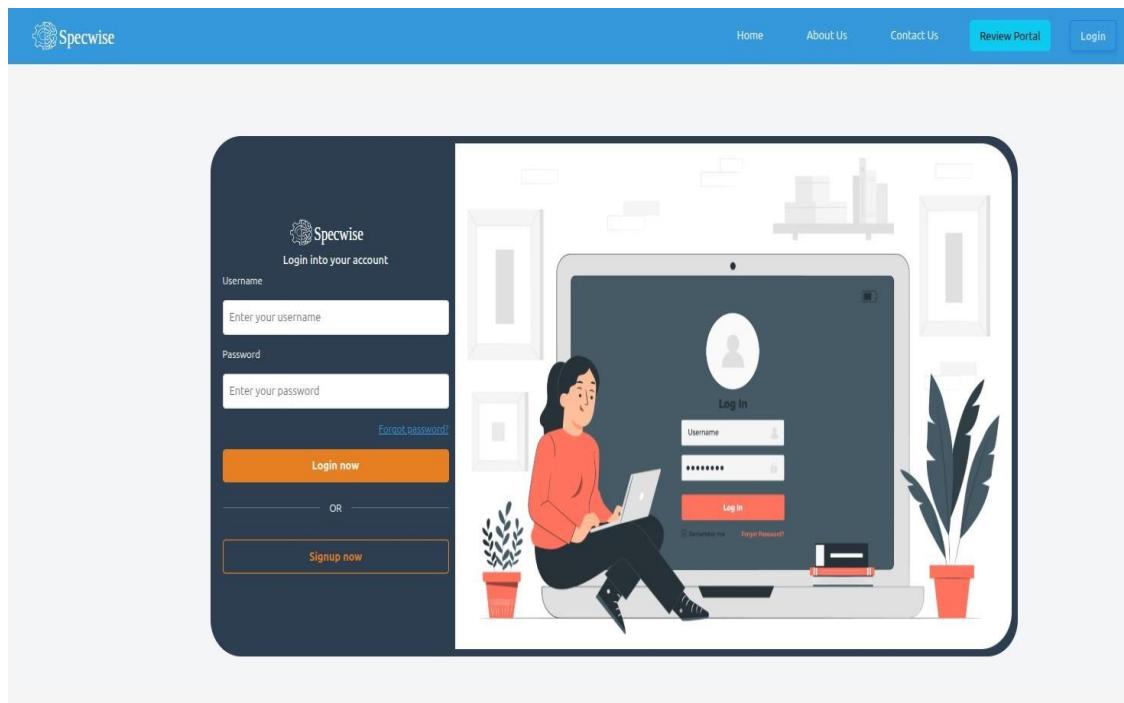


Figure 12: Specwise Login page

Returning users can securely sign in on this page. It's simple, quick and straightforward.

5.2.3 Sign up Page

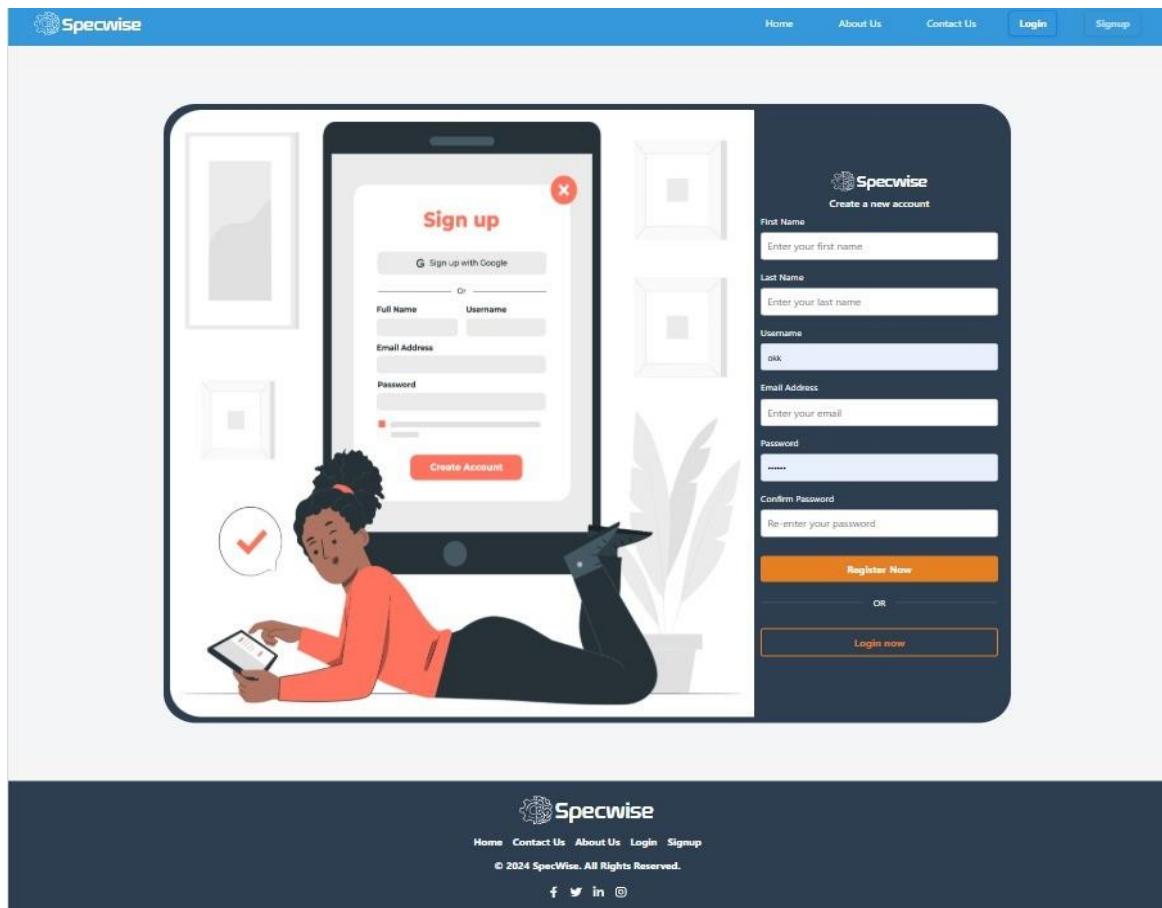


Figure 13: Specwise Signup Page

Here, new users can easily create an account by entering their details. The process is straightforward and user-friendly, helping you join SpecWise in just a few clicks.

5.2.3 About us

The screenshot shows the 'About Us' section of the Specwise website. At the top, there's a blue header bar with the 'About Us' title. Below it is a large white area containing a sub-section titled 'Our Mission'. This section features a central illustration of a person interacting with a large brain surrounded by circuit board patterns and various icons like a lightbulb, gears, and a chess piece. To the left of the illustration is a text block about the company's mission. Below this is a section titled 'Our Values' with three cards: 'Innovation' (represented by a lightbulb icon), 'User-Centricity' (represented by a user profile icon), and 'Reliability' (represented by a shield icon). Each card has a brief description of the value.

About Us

Learn more about Specwise and our mission to help you choose the perfect PC hardware for your needs.

Our Mission

At Specwise, our mission is to make hardware selection easy and efficient for everyone. Whether you're a gamer, content creator, or professional, our dedicated team works hard to deliver the best tools and resources to help you choose the perfect hardware for your specific needs. We ensure that your PC meets the performance demands of your workload with a tailored approach that puts you first.

Innovation

We continuously seek to improve and innovate to make hardware selection easy for our users.

User-Centricity

Our platform is designed specifically to deliver user-friendly solutions that match your needs.

Reliability

We are committed to providing accurate and trusted guidance for all hardware-related decisions.

Figure 14: Specwise About Us Page

The About Us page shares the story behind SpecWise, introducing the team and our mission to make PC building easy and accessible for everyone.

5.2.4 Contact us

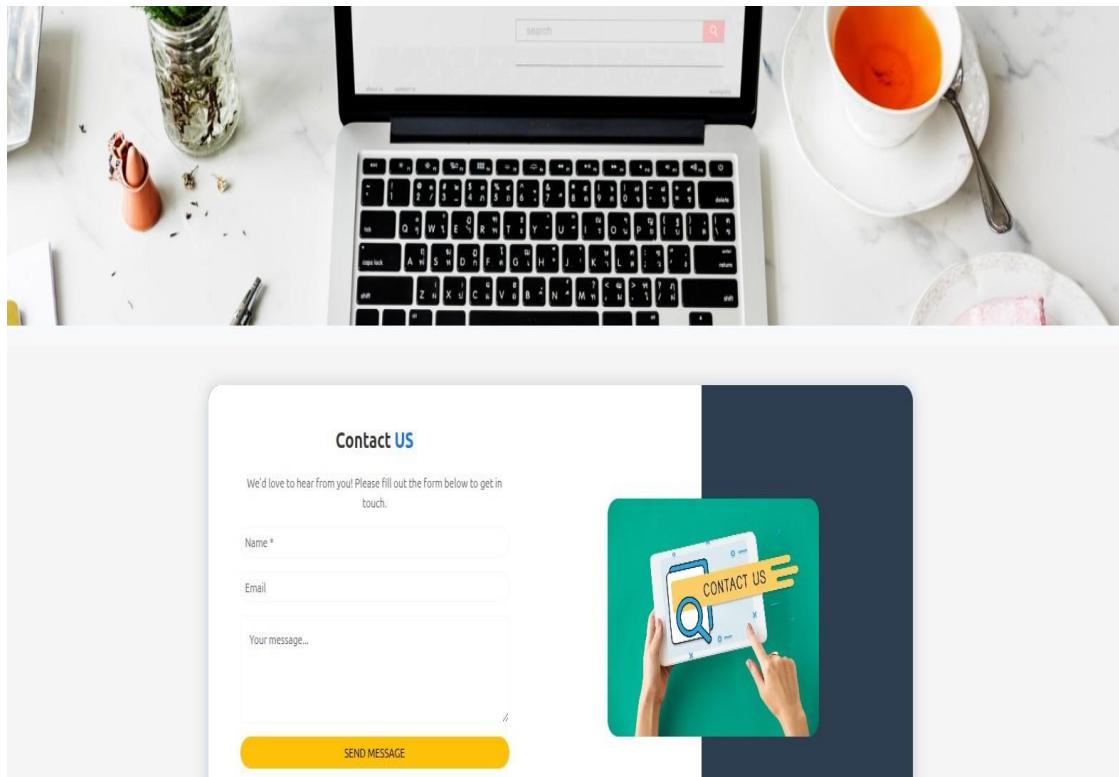


Figure 15: Specwise Contact Page

On the Contact Us page, users can quickly reach out with questions, feedback, or support requests. It's designed to make connecting with our team simple and hassle-free.

5.2.5 Reviews Page

The screenshot shows the Specwise Review Portal. At the top, there is a blue header bar with the Specwise logo on the left and navigation links for Home, About Us, Contact Us, Review Portal (which is highlighted in blue), Login, and Signup on the right. Below the header, the title "User Review Portal" is displayed in a yellow starburst font. A dropdown menu labeled "Sort by Date" is visible. The main content area contains three review cards, each with a "Question" section and a "Rating" section. The first review asks about running GTA 5 and building a PC, has a 0/5 rating, and was reviewed on 4/16/2025. The second review asks for HD graphics design software, has a 2/5 rating, and was reviewed on 4/11/2025. The third review asks for recommendations for graphic designing, has a 4/5 rating, and was updated on 4/11/2025.

Question	Rating	Review	Reviewed on
i want to run gta 5 and im building a... See More	★☆☆☆☆ (0/5)	Review	4/16/2025
suggest me the HD for graphic designig.	★☆☆☆☆ (2/5)	Review vgtucvyr	4/11/2025
recommend me ram to graphic designing.	★★★★☆ (4/5)	Review Updated my review — It's actually very useful!	4/11/2025

Figure 16: Specwise Review page

The Reviews page lets users read and share honest feedback about their experiences with SpecWise and its recommendations. It's a space for the community to connect, offer insights, and help others make informed choices.

5.2.6 Review History Page

The screenshot shows the Specwise Review History Page. At the top, there is a navigation bar with links for Home, About Us, Contact Us, Review Portal, Review History (which is highlighted in blue), History, Welcome, shah, and Logout. Below the navigation bar, the page title is "Your Review History" with two yellow stars on either side. A dropdown menu labeled "All Reviews" is open. Two review items are listed in cards:

- Review 1:** "i want to run gta 5 and im building a new computer so i want all the hardware" - Rating: 0 / 5. The review text is bolded. Below it is a "Review:" section which is currently empty. The date is listed as "Date: 4/16/2025, 12:30:11 AM".
- Review 2:** "i want to run blender give me the best gpu for it" - Rating: 0 / 5. The review text is bolded. Below it is a "Review:" section which is currently empty. The date is listed as "Date: 1/1/1970, 5:00:00 AM".

A blue "Write a Review" button is located at the bottom right of the card for the second review.

Figure 17: Specwise Review History Page

The Review History page keeps a private log of all your past questions and requests, so you can easily track, review, or revisit your previous interactions with SpecWise. It's a convenient way for logged-in users to stay organized and never lose their progress.

5.2.7 Review History Panel

Question ID	Question	Timestamp	Action
168	i want to run blender	5/25/2025, 8:22:32 PM	<button>View Details</button>
165	i want to run blender give me the best gpu for it	4/15/2025, 10:47:08 PM	<button>View Details</button>
163	i want to run gta 5 and im building a new computer so i want all the hardware	4/15/2025, 5:00:30 PM	<button>View Details</button>

Figure 18: Specwise Recommendation History Page

Recommendation Details

Question: i want to run blender give me the best gpu for it
Time: 4/15/2025, 10:47:08 PM

Software: Blender

Task: 3D modeling and animation

Hardware: GPU

Suitable Hardware

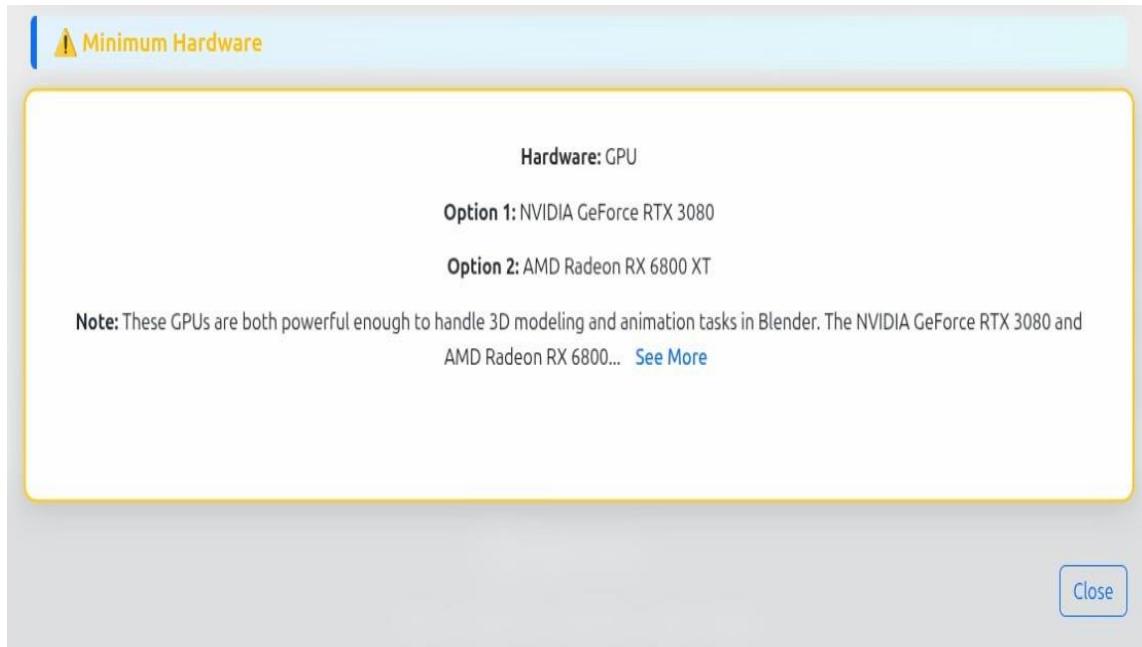
Hardware: GPU

Option 1: NVIDIA GeForce RTX 3080

Option 2: AMD Radeon RX 6800 XT

Note: These two GPUs are currently among the best options for 3D modeling and animation due to their high performance and VRAM capacity. The NVIDIA GeForce... [See More](#)

Figure 19: Specwise Recommendation Page Details



Recommendation Page Details

The History panel keeps a detailed record of all your previous questions along with the exact hardware recommendations provided for each one. It's your personal archive, making it easy to revisit past queries, review what was suggested, and track how your needs or our advice have changed over time.

5.2.8 Recommendation page

The screenshot shows the Specwise recommendation page. At the top, there is a navigation bar with links for Home, About Us, Contact Us, Review Portal, Review History, History, Welcome, shah, and Logout. The main heading is "WELCOME TO SPECWISE" with the subtext "LET'S FIND HARDWARE FOR YOU". Below this, a subtext says "Find the perfect PC hardware based on your software and usage needs." A large central box contains a "Tell Us Your Requirements" section with a text area for input. The input text reads: "I want to run Adobe Photoshop 2023, Autodesk AutoCAD 2024, and Visual Studio 2022 and also do image processing for large-scale projects. I'll be working with high-resolution graphics and need fast rendering speeds. What PC hardware should I use?". Below this is a "Get Recommendations" button. Below the input box is a "Question Details" section with a summary of the input question: "Input Question: i want to run blender". This is followed by four boxes: "Software Name" (Blender), "Task to Perform" (3D modeling and animation), "Hardware Mentioned" (PC), and "Special Task" (None). The next section is titled "Minimum Hardware Requirements" and includes two tables: one for "CPU" and one for "GPU". The CPU table has columns for Requirement, Option 1, Option 2, and Note. The note for the minimum requirement states: "These CPUs are both powerful options for 3D modeli..." and has a "See More" link. The GPU table has columns for Requirement, Option 1, Option 2, and Note. The note for the minimum requirement states: "These GPUs are suitable for 3D modeling and animat..." and has a "See More" link.

WELCOME TO SPECWISE
LET'S FIND HARDWARE FOR YOU
Find the perfect PC hardware based on your software and usage needs.

Tell Us Your Requirements
Please provide the details of the software and tasks you'll be performing, and we'll recommend the best hardware configuration.

Please enter your Question

I want to run Adobe Photoshop 2023, Autodesk AutoCAD 2024, and Visual Studio 2022 and also do image processing for large-scale projects. I'll be working with high-resolution graphics and need fast rendering speeds. What PC hardware should I use?

Get Recommendations

Question Details

Input Question: i want to run blender

Software Name Blender	Task to Perform 3D modeling and animation
Hardware Mentioned PC	Special Task None

Minimum Hardware Requirements

CPU

Requirement	Option 1	Option 2	Note
Minimum	AMD Ryzen 7 5800X	Intel Core i9-10900K	These CPUs are both powerful options for 3D modeli... See More

GPU

Requirement	Option 1	Option 2	Note
Minimum	NVIDIA GeForce RTX 3060 Ti	NVIDIA Quadro RTX 5000	These GPUs are suitable for 3D modeling and animat... See More

*Figure 21:
Specwise Recommendation
Page*

Hardware Product Details:				
Name	Category	Price	Rating	Link
AMD 100 10000031SBX Ryzen 5 3600	CPU	\$88.99	4.9	View Product
AMD Ryzen 9 5900X 3.7 GHz 12-Core AM4 Processor	CPU	\$549.00	5	View Product
AMD Ryzen 9 5900XT 3.3 GHz 16-Core AM4 Processor	CPU	\$323.99	5	View Product
Intel Core i9-14900K 3.2 GHz 24-Core LGA 1700 Processor	CPU	\$449.99	4.5	View Product
Intel Core i9-12900K 3.2 GHz 16-Core LGA 1700 Processor	CPU	\$378.00	5	View Product
Intel Core i9-13900KS 3.2 GHz 24-Core LGA 1700 Processor	CPU	\$499.99	4.5	View Product
NVIDIA RTX A4000 Graphics Card	GPU	\$1,109.00	N/A	View Product
PNY NVIDIA RTX A400 Graphics Card	GPU	\$249.00	N/A	View Product
PNY NVIDIA RTX A6000 Graphics Card	GPU	\$5,299.99	3.5	View Product
NVIDIA RTX A4000 Graphics Card	GPU	\$1,109.00	N/A	View Product

Recommendation page users can simply enter their question or describe your needs, and our AI analyzes their input for software, tasks, and any special requirements. It then presents them with the minimum hardware specs and a handy table of recommended parts, each with direct links for easy browsing and purchase.

5.2.9 Admin Panel

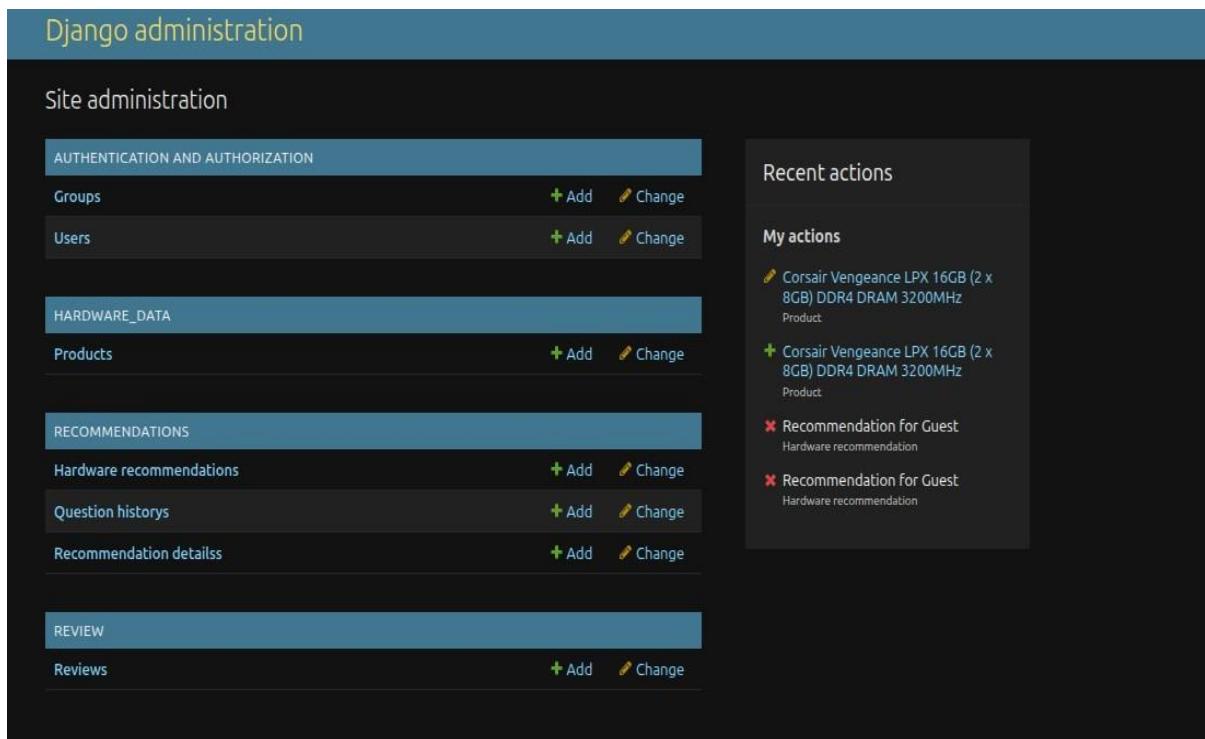


Figure 22: Specwise Admin Panel

The Admin Panel gives administrators full access to all databases, making it easy to view, edit, and manage platform data as needed. It's a powerful tool for keeping everything organized and up to date.

5.2.10 Review Panel

The screenshot shows a web-based admin panel for managing reviews. At the top, there's a search bar with a magnifying glass icon and a 'Search' button. To the right are two buttons: 'ADD REVIEW +' and 'SEND REMINDER +'. On the far right, there's a 'FILTER' section with several dropdown menus and buttons.

The main area displays a table of reviews. The columns are: Action, USER, QUESTION, REVIEW RATING, REVIEW STATUS, REVIEW DATE, and CURRENT DATE. There are 10 reviews listed:

Action	USER	QUESTION	REVIEW RATING	REVIEW STATUS	REVIEW DATE	CURRENT DATE
<input type="checkbox"/>	shah	shah - i want to run blender	0	X	-	May 25, 2025, 3:22 p.
<input type="checkbox"/>	shahbaz	shahbaz - I want to run blender give me the best gpu for it	0	X	-	April 16, 2025, 7:28 a
<input type="checkbox"/>	shahbaz	shahbaz - i want to know the capital of pakistan	0	X	-	April 16, 2025, 7:28 a
<input type="checkbox"/>	shah	shah - i want to run blender give me the best gpu for it	0	X	-	April 15, 2025, 5:47 p
<input type="checkbox"/>	johndoe	johndoe - What graphics card is best for Blender rendering?	0	X	-	April 15, 2025, 5:35 p
<input type="checkbox"/>	shah	shah - i want to run gta 5 and im building a new computer	0	✓	April 15, 2025, 7:38 p.m.	April 15, 2025, noon
<input type="checkbox"/>	-	Guest - i want to run gta 5 give me the complete setup	0	X	-	April 14, 2025, 9:47 a
<input type="checkbox"/>	-	Guest - i want to run gta 5	0	X	-	April 14, 2025, 9:46 a
<input type="checkbox"/>	ts	ts - What is the best GPU for gaming under \$300?	0	X	-	April 14, 2025, 9:44 a
<input type="checkbox"/>	okk	okk - suggest me ram for Natural language processing.	0	X	-	April 14, 2025, 8:24 a

At the bottom, there are page navigation buttons (1, 2, 3, 4) and a link to '37 reviews Show all'.

Figure 23: Specwise Review Admin Panel

In the Review Panel, admins can see all user-submitted reviews and quickly reach out to users via email to remind them about any pending feedback. This helps ensure reviews are timely and the community stays active.

5.2.11 User History Panel

The screenshot shows the Specwise Question History Panel. At the top, there is a search bar with a magnifying glass icon and a 'Search' button. To the right of the search bar is a 'FILTER' section with a 'Show counts' checkbox and a dropdown menu set to 'By is hardware'. Below these are three filter options: 'All', 'Yes', and 'No'. The main area displays a table of questions. The columns are 'USER', 'QUESTION', 'IS HARDWARE', and 'TIMESTAMP'. There are 10 selected items out of 95 total. The questions listed include various queries about hardware and software, such as 'i want to run blender', 'What graphics card is best for Blender rendering?', and 'What is the best GPU for gaming under \$300?'. The 'IS HARDWARE' column shows green checkmarks for most questions, indicating they are hardware-related. The 'TIMESTAMP' column shows dates ranging from April 14, 2025, to May 25, 2025. At the bottom of the table, there is a page navigation bar with numbers 1 through 10 and a 'Show all' link.

USER	QUESTION	IS HARDWARE	TIMESTAMP
shah	i want to run blender	✓	May 25, 2025, 3:22 p.m.
shahbaz	I want to run blender give me the best gpu for it	✓	April 16, 2025, 7:28 a.m.
shahbaz	i want to know the capital of pakistan	✗	April 16, 2025, 7:28 a.m.
shah	i want to run blender give me the best gpu for it	✓	April 15, 2025, 5:47 p.m.
johndoe	What graphics card is best for Blender rendering?	✓	April 15, 2025, 5:35 p.m.
shah	i want to run gta 5 and im building a new computer so i want all the hardware	✓	April 15, 2025, noon
-	i want to run gta 5 give me the complete setup	✓	April 14, 2025, 9:47 a.m.
-	i want to run gta 5	✓	April 14, 2025, 9:46 a.m.
ts	What is the best GPU for gaming under \$300?	✓	April 14, 2025, 9:44 a.m.
okk	suggest me ram for Natural language processing.	✓	April 14, 2025, 8:24 a.m.

Figure 24: Specwise Question History Panel

The Questions History panel keeps a handy record of all the questions you've asked on SpecWise. It makes it easy to revisit, review, or follow up on your past queries whenever you need.

5.2.12 Products Panel

Select product to change						ADD PRODUCT +
Action:	ID	NAME	CATEGORY	PRICE	RATING	LINK
<input type="checkbox"/>	5328	PCIe 6Pin to 4 SATA 15Pin Male SSD Power Supply Cable For Seasonic PSU Modular X Series X-1050XM2,X-1250XM2, X-650 KM3, X-750 KM3, X-850 KM3 18AWG	PowerSupplies	\$10.22	-	https://www.newegg.com/p/2S7-09KG-006T1
<input type="checkbox"/>	5327	PCIe 6Pin to 4 SATA 15Pin Male SSD Power Supply Cable For Seasonic PSU Modular X Series X-1050XM2,X-1250XM2, X-650 KM3, X-750 KM3, X-850 KM3 18AWG	PowerSupplies	\$9.22	-	https://www.newegg.com/p/2S7-09KG-006U3
<input type="checkbox"/>	5326	PCI express 6Pin to 3 SATA 15Pin Power Supply Cable Port Multiplier for CORSAIR RM1000 RM850 RM750 RM650 RM550 RM450 Modular PSU	PowerSupplies	\$9.19	-	https://www.newegg.com/p/2S7-09KG-005P4
<input type="checkbox"/>	5325	30V 10A Adjustable DC Power Supply Precision Variable Dual Digital Lab Test 110V	PowerSupplies	\$119.99	-	https://www.newegg.com/p/1HU-028C-00041
<input type="checkbox"/>	5324	YEECHUN 265W New Replacement Power Supply Compatible with Dell Optiplex 390 3010 790 990 MT Mini Tower Compatible Part Numbers: L265EM-00 F265EM-00 AC265AM-00 H265AM-00 YC7TR 9D9T1 GVY79 053N4 D3D1C	PowerSupplies	\$59.99	-	https://www.newegg.com/p/1HU-02JX-006K7
<input type="checkbox"/>	5323	Slimline Power Supply Upgrade for SFF Desktop Computer - Fits: Enhance ENP-2220A, ENP-2222A, ENP-2222B, ENP-2224B,	PowerSupplies	\$69.99	-	https://www.newegg.com/replace-power-psu-sff-enhance/p/1HU-009N-00083
<input type="checkbox"/>	5322	300W Mute Power Supply Suitable for ASIC miner Goldshell Mini-DOGE KD-BOX HS-BOX LB-BOX CK-BOX	PowerSupplies	\$94.89	-	https://www.newegg.com/p/1HU-033E-00005
<input type="checkbox"/>	5321	Power Supply Replacement for Dell OK159T CX305P-00 CX305P-00 D350R003L	PowerSupplies	\$79.98	-	https://www.newegg.com/p/1HU-027T-000F5
<input type="checkbox"/>	5320	Replace Power Supply TFX for HP Pavilion s5250t CTO s5257c s5260f	PowerSupplies	\$69.98	-	https://www.newegg.com/p/1HU-009N-00660

Figure 25: Specwise Products Panel

The Products panel lets you browse and manage all the hardware items available on SpecWise. It's your go-to spot for viewing details, updating listings, or adding new products to keep our recommendations fresh and accurate.

5.2.13 Recommendation Details Panel

Select recommendation details to change

Action: ----- Go 0 of 10 selected

QUESTION HISTORY	SOFTWARE NAME	TASK TO PERFORM	HARDWARE MENTIONED	SPECIAL TASK
shah - i want to run blender	Blender	3D modeling and animation	PC	None
shahbaz - I want to run blender give me the best gpu for it	Blender	3D modeling and animation	GPU	None (but it's implied that the GPU should be powerful enough to handle 3D modeling and animation)
shah - i want to run blender give me the best gpu for it	Blender	3D modeling and animation	GPU	None (but it is implied that the GPU should be powerful enough to handle 3D modeling and animation)
shah - i want to run gta 5 and im building a new computer	Grand Theft Auto V (GTA 5)	Gaming	PC	None (but the user is building a new computer for gaming, so it's assumed that they want decent performance for gaming)
Guest - i want to run gta 5 give me the complete setup	Grand Theft Auto V (GTA 5)	Gaming	GPU, CPU, RAM, Motherboard (implied), PC (implied)	High-resolution graphics, fast rendering speeds (implied)

FILTER

Show counts

By software name

All

- Adobe Creative Suite (Photoshop, Illustrator, InDesign, etc.)
- Adobe Creative Suite (assuming graphic designing software is Adobe-based)
- Adobe Creative Suite (assuming it's a general graphic design software)
- Adobe Creative Suite (or any other graphic design software)
- Adobe Photoshop, Adobe Illustrator, CorelDRAW, Autodesk Sketchbook, etc. (Since graphic designing can be done on various software, I've listed some popular ones.)
- Adobe Photoshop, Adobe Illustrator, CorelDRAW, Sketch, etc. (Since the question is about graphic designing, it's likely that the user will be using one of these software or similar graphic design software.)
- Adobe Photoshop, Adobe Illustrator, CorelDRAW, or any other graphic design software.

Figure 26: Specwise Recommendation Details Panel

The Recommendations Detail panel breaks down each user question into its key components such as software, tasks, and any special requirements and stores them in the database. This organized view helps us track exactly how recommendations are made and makes it easier to analyze or refine our AI's logic in the future.

5.2.14 Web scraping Panel

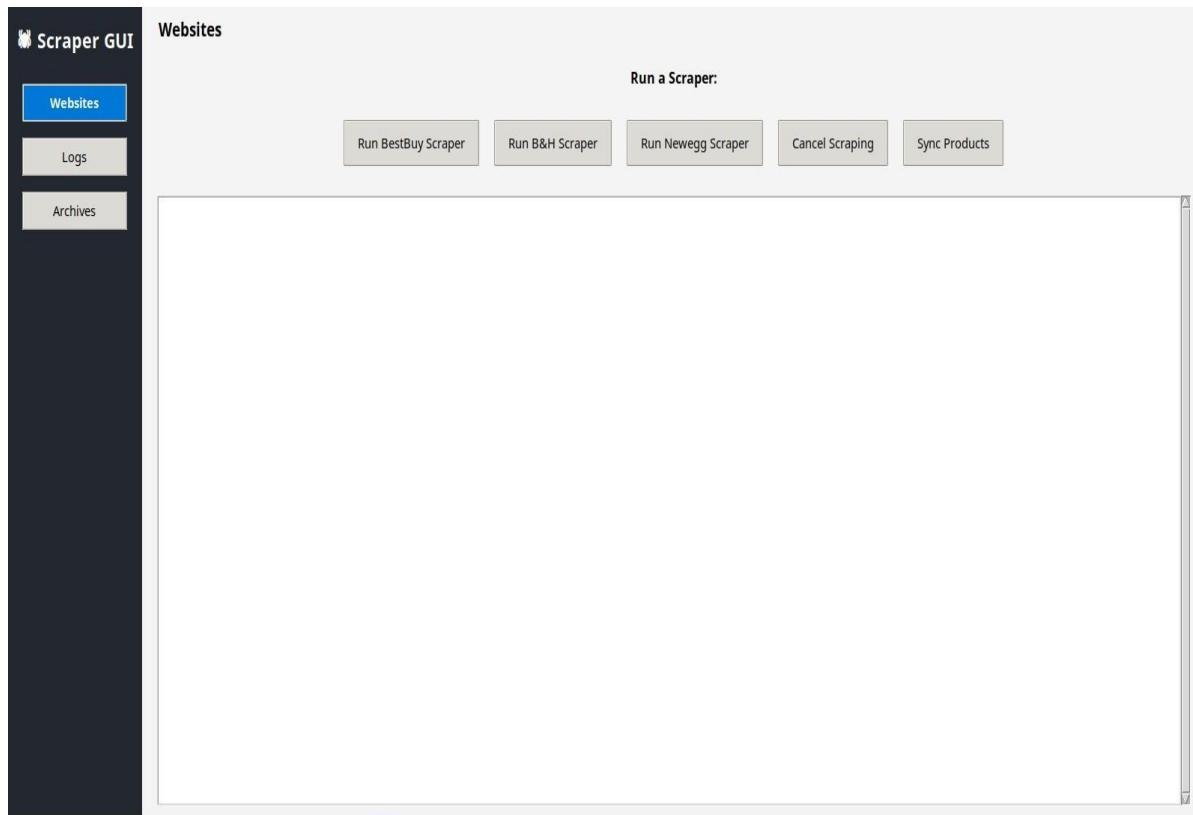


Figure 27: Specwise web scraping Gui

The web scraping GUI is an all-in-one control panel for managing and monitoring the entire scraping process. With a clean sidebar for easy navigation, we can quickly launch scrapers for different websites, view real-time logs, and access archived data all from one place.

5.2.15 Web scraping Logs Panel

Logs									
View Logs:									
BestBuy Logs B&H Logs Newegg Logs									
Filter by Date: <input type="button" value="2025-04-06"/>									
Date	Datetime	Product	Old Reviews	New Reviews	Old Price	New Price	Old Rating	New Rating	
2025-04-06	2025-04-06 01:50:59	Intel - Core i9-14900K 14th Gen	249	293	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:50:59	AMD - Ryzen 5 7600X 6-core - 12	198	232	\$209.00	\$219.00	nan	nan	
2025-04-06	2025-04-06 01:50:59	AMD - Ryzen 7 7800X3D 8-Core	721	785	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:50:59	Intel - Core i7-14700K 14th Gen	168	198	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:51:00	AMD - Ryzen 9 7900X 12-core - 2	94	130	\$369.00	\$374.00	nan	nan	
2025-04-06	2025-04-06 01:51:00	AMD - Ryzen 7 7700X 8-core - 16	76	88	\$309.00	\$319.00	nan	nan	
2025-04-06	2025-04-06 01:51:00	Intel - Core i5-14600K 14th Gen	11	15	nan	nan	4.6	4.7	
2025-04-06	2025-04-06 01:51:00	AMD - Ryzen 9 7950X 16-core - 3	146.0	153	\$519.00	\$549.00	nan	nan	
2025-04-06	2025-04-06 01:51:23	Package - AMD - Ryzen 7 7700X	nan	nan	\$508.99	\$518.99	nan	nan	
2025-04-06	2025-04-06 01:51:23	High End Build Your Own Gaming	nan	nan	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:51:23	Mid-Range Build Your Own Gami	nan	nan	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:51:23	Mid-Range Build Your Own Gami	nan	nan	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:51:23	Entry Level Build Your Own Gam	nan	nan	nan	nan	nan	nan	
2025-04-06	2025-04-06 01:51:23	Intel - Core i7-13700K 13th Gen	445	455	\$379.99	\$314.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	XFX - SPEEDSTER SWFT210 AMD	None	20	\$259.99	\$319.99	None	4.5	
2025-04-06	2025-04-06 01:52:18	XFX - Speedster MERC310 AMD I	106.0	181	\$669.99	\$869.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	XFX - SPEEDSTER MERC319 AMD	326.0	414	\$489.99	\$629.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	MSI - NVIDIA RTX 3050 GAMING	18.0	50	\$179.99	\$204.99	4.7	4.6	
2025-04-06	2025-04-06 01:52:18	XFX - SPEEDSTER QICK319 AMD I	1.0	42	\$389.99	\$529.99	5.0	4.9	
2025-04-06	2025-04-06 01:52:18	XFX - SPEEDSTER SWFT210 AMD	573.0	587	\$209.99	\$265.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	XFX - SPEEDSTER SWFT210 AMD	393.0	430	\$234.99	\$299.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	MSI - NVIDIA RTX 3050 VENTUS	139.0	168	\$219.99	\$224.99	nan	nan	
2025-04-06	2025-04-06 01:52:18	MSI - NVIDIA RTX 3050 VENTUS	544.0	550	\$169.99	\$224.99	nan	nan	

Figure 28: Specwise Web scraping Logs Panel

The Web scraping Logs panel provides a clear, searchable record of all scraping activities, updates, and changes. It's designed to make monitoring, troubleshooting, and reviewing the scraping process straightforward, with filters to quickly find logs by date or website.

5.2.16 Web scraping Archive Panel

Archived Files:								
category	last_updated	link	name	price	rating	reviews	store_id	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i5-14600H	\$259.99	4.6	10.0	1	
CPUUs	2024-12-15 19:51:28	https://www.bestbuy.com	AMD Ryzen 9 5900X	\$269.00	4.9	854.0	1	
CPUUs	2024-12-15 19:51:29	https://www.bestbuy.com	AMD Ryzen 5 5600X	\$124.00	4.9	1688.0	1	
CPUUs	2025-04-06 01:51:00	https://www.bestbuy.com	AMD Ryzen 9 7900X	\$374.00	4.9	130.0	1	
CPUUs	2025-04-06 01:50:59	https://www.bestbuy.com	Intel Core i9-14900H	\$499.99	4.4	293.0	1	
CPUUs	2025-04-06 01:50:59	https://www.bestbuy.com	AMD Ryzen 5 7600X	\$219.00	4.8	232.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i7-14700H	\$399.99	4.6	166.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i7-14700H	\$399.99	4.6	166.0	1	
CPUUs	2025-04-06 01:51:00	https://www.bestbuy.com	AMD Ryzen 7 7700X	\$319.00	4.9	88.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i5-14600H	\$259.99	4.6	10.0	1	
CPUUs	2024-12-15 19:51:29	https://www.bestbuy.com	AMD Ryzen 7 9700X	\$359.00	4.8	16.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 7 5800XT	\$239.00	3.3	6.0	1	
CPUUs	2025-04-06 01:51:00	https://www.bestbuy.com	AMD Ryzen 9 7950X	\$549.00	4.7	153.0	1	
CPUUs	2024-12-15 19:51:30	https://www.bestbuy.com	AMD Ryzen 5 5500	\$87.00	4.7	73.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 5 9600X	\$279.00	4.0	2.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 9 9900X	\$499.99	5.0	2.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 5 4500	\$69.00	4.8	51.0	1	
CPUUs	2024-12-15 19:51:30	https://www.bestbuy.com	Intel Core Ultra 7 26E	\$429.99	4.3	3.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 7 5700	\$139.00	1.0	1.0	1	
CPUUs	2024-12-15 19:51:30	https://www.bestbuy.com	AMD Ryzen 9 5900XT	\$349.00	5.0	2.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i5-14600H	\$259.99	4.6	10.0	1	
CPUUs	2025-04-06 01:51:23	https://www.bestbuy.com	Intel Core i7-13700H	\$314.99	4.8	455.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Package - AMD - AMD	\$718.99	nan	nan	1	
CPUUs	2024-12-15 19:52:05	https://www.bestbuy.com	AMD Ryzen 7 5800X	\$449.00	4.8	1366.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 5 8500C	\$149.00	4.0	1.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Package - AMD - Ryze	\$508.99	nan	nan	1	
CPUUs	2025-04-06 01:50:59	https://www.bestbuy.com	Intel Core i7-14700H	\$399.99	4.6	198.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	AMD Ryzen 5 7600	\$214.00	4.5	28.0	1	
CPUUs	2025-04-15 15:25:30	https://www.bestbuy.com	Intel Core i5-13600H	\$225.99	4.9	233.0	1	
CPUUs	2025-04-06 01:51:23	https://www.bestbuy.com	Entry Level Build You	\$1,211.92	nan	nan	1	
CPUUs	2024-12-15 19:52:06	https://www.bestbuy.com	Entry Level Build You	\$895.93	nan	nan	1	
CPUUs	2025-04-06 01:51:23	https://www.bestbuy.com	Mid-Range Build Your	\$1,699.93	nan	nan	1	

Figure 29: Specwise Web scraping Archives Panel

The Web scraping Archives panel gives you easy access to all saved data files from previous scrapes. You can quickly browse, open, and sort through CSV archives in a familiar table format, making it simple to review past results or compare historical data.

6 Algorithm analysis and complexity

6.1 *Web Scraping*

Algorithms were required to efficiently scrape hardware data from e-commerce platforms like BestBuy, B&H and Newegg for variables that were product names, prices, ratings, and reviews.

6.1.1 Page Traversal and Data Extraction:

Algorithm: Depth-First Search (DFS)

Iteratively or recursively visits pages to extract hardware information while handling paginated results.

Pseudocode:

Input:

category_url: URL of the product category
max_pages: Maximum number of pages to scrape

Output:

List of extracted product data

Initialize page_num = 1

While page_num <= max_pages:

 Construct URL with page_num

 Load URL and wait for elements to appear

 For each product on the page:

 Extract name, price, rating, reviews, link

 Append extracted data to results list

 Increment page_num

Return product data

6.1.2 Human-like Scrolling:

Algorithm: Iterative Scrolling Simulation:

Scrolls through pages to load content dynamically and introduces delays between actions emulate human behavior to avoid detection.

Pseudocode:

```
Initialize last_height = current_page_height
While new_height != last_height:
    Scroll to bottom of the page
    Wait random delay (to simulate user behavior)
    Update last_height with new_height
```

6.1.3 Data Deduplication and Change Detection:

Algorithm: Hashing:

It uses a hash table or set to identify unique products and detect changes in prices or ratings and it also ensures that only new or updated data is saved.

Pseudocode:

```
Input:
    existing_data: Current database of products
    new_data: Newly scraped data
```

Output:

Updated database with changes logged

For each product in new_data:

```
If product is in existing_data:
    Compare fields (price, rating, reviews)
```

```

If changes are detected:
    Update existing_data and log changes
Else:
    Add product to existing_data
Return updated database

```

6.1.4 Complexity Analysis

Algorithm	Purpose	Time Complexity	Space Complexity
Page Traversal (DFS)	Scraping paginated product data	$O(P \cdot E)$	$O(E)$
Scrolling Simulation	Load dynamic content	$O(S)$	$O(1)$
Deduplication	Avoid duplicate records	$O(N)$	$O(N)$

Table 3: Webscrape Complexity Analysis

6.2 LLM Prompt System

6.2.1 Prompt classification

Algorithm:

Categorize user queries into "Recommendation" or "Sufficiency."

Pseudocode:

Input: question

Output: classification

template = "'''"

You are a PC hardware recommendation classifier.

Answer 'Yes' or 'No' if the question explicitly requests PC hardware recommendations.

Question: {question}

Answer:

"""

```
response = LLM.invoke(template.format(question=question))
```

```
classification = response.strip().lower()
```

```
return classification
```

6.2.2 Recommendation Extraction

Algorithm:

Extract details (e.g., Software Name, Task to Perform, Hardware) from user queries.

Pseudocode:

Input: question

Output: extracted_details

template = """"

You are an expert in PC hardware analysis.

Extract the following details:

1. Software Name
2. Task to Perform
3. Hardware Mentioned
4. Special Task

Question: {question}

Response:

"""

```
response = LLM.invoke(template.format(question=question))
```

```
software_name = regex_extract(response, "Software Name: (.*)? Task to Perform")
```

```
task_to_perform = regex_extract(response, "Task to Perform: (.*)? Hardware  
Mentioned")
```

```
hardware_mentioned = regex_extract(response, "Hardware Mentioned: (.*)? Special  
Task")
```

```

special_task = regex_extract(response, "Special Task: (.*)")
extracted_details = {
    "Software Name": clean(software_name),
    "Task to Perform": clean(task_to_perform),
    "Hardware Mentioned": clean(hardware_mentioned),
    "Special Task": clean(special_task)
}

return extracted_details

```

6.2.3 Hardware Recommendation

Algorithm:

Generate hardware recommendations based on extracted user requirements.

Pseudocode:

Input: extracted_details

Output: llm_recommendations

```

software_name = extracted_details["Software Name"]
task_to_perform = extracted_details["Task to Perform"]
special_task = extracted_details["Special Task"]
hardware_types = extracted_details["Hardware Mentioned"].split(", ")

if hardware_types == ["PC"]:
    hardware_types = ["CPU", "GPU", "RAM", "SSD", "Motherboard", "Power
Supply"]

llm_recommendations = {}

```

for hardware in hardware_types:

```
template = """
```

Based on the details:

- Software Name: {software_name}
- Task to Perform: {task_to_perform}
- Special Task: {special_task}

Recommend 2 {hardware} options:

```
"""
```

```
response = LLM.invoke(template.format(
```

```
    software_name=software_name,  
    task_to_perform=task_to_perform,  
    special_task=special_task,  
    hardware=hardware
```

```
))
```

```
llm_recommendations[hardware] = parse_llm_output(response)
```

```
return llm_recommendations
```

6.2.4 Complexity Analysis

Algorithm	Purpose	Time Complexity	Space Complexity
Classification	Categorize user queries	$O(N)$	$O(N)$
Extraction	Extract structured data	$O(N)+O(M)$	$O(N+M)$
Recommendation	Generate hardware suggestions	$O(N \cdot C)+O(M)$	$O(N \cdot C+M \cdot C)$

Table 4: LLM Prompt Complexity Analysis

The complexity tables above provide a clear summary of the computational resources required by each algorithm in both the web scraping and LLM prompt systems. By analyzing the time and space complexities, we ensure that each component of the system remains efficient and scalable as the volume of data and number of users grow. These insights are crucial for anticipating potential bottlenecks and for guiding future optimizations.

It is important to note that, while the theoretical complexities offer valuable guidance, real-world performance may also be influenced by external factors such as network latency, server response times, and the dynamic behavior of target websites. As such, ongoing monitoring and empirical testing are essential to validate and refine these complexity estimates in practical deployments.

7 Implementation

7.1 Hardware Recommendation System

One of the most important and unique features of SpecWise is the use of a Large Language Model (LLM) to generate tailored hardware recommendations based on the user's input.

7.1.1 LLM-Based Prompt Processing

Once a user enters their software requirements or task (e.g., “I want to edit videos using Adobe Premiere Pro”), the system uses the “extract_recommendation_details()” function to break the input into four categories:

- Software Name
- Task to Perform
- Hardware Mentioned

The extraction is done using a prompt sent to the Hugging Face API, which uses the Mistral-7B language model. After receiving the structured response, the fields are parsed using regular expressions and cleaned for further processing.

Code:

```
def extract_recommendation_details(question):

    prompt = PromptTemplate(template=..., input_variables=["question"])

    llm_chain = LLMChain(prompt=prompt, llm=llm)

    response = llm_chain.invoke({"question": question})

    # extract fields from response using regex
```

Output:

```
{  
    "Software Name": "Adobe Premiere Pro",  
    "Task to Perform": "Video editing",  
    "Hardware Mentioned": "GPU, CPU",  
    "Special Task": "High-resolution rendering"  
}
```

7.1.2 Hardware Generation using LLM

After extracting the required information, the function “recommend_hardware_with_llm()” is called. This function generates two types of recommendations for each component:

- Minimum Requirements
- Suitable (Recommended) Hardware

For each hardware type (CPU, GPU, etc.), a templated prompt is dynamically filled and sent to the LLM. The model returns two hardware options along with a short note or comment.

Code:

```
def recommend.hardware.with.llm(extracted_details, recommendation_type):  
    for hardware in hardware_types:  
        prompt = PromptTemplate(template=..., input_variables=[...])  
        llm_chain = LLMChain(prompt=prompt, llm=llm)  
        response = llm_chain.invoke({...})  
        # parse and store options
```

The responses are parsed and formatted using the `parse_llm_output()` utility, which cleans up the model's output and separates it into:

- option 1
- option 2
- note

Output:

```
"GPU": {  
    "option1": "NVIDIA RTX 3060",  
    "option2": "AMD Radeon RX 6700 XT",  
    "note": "Both are suitable for smooth 4K editing."  
}
```

7.1.3 Parallel Recommendation Fetching

To reduce response time and optimize performance, SpecWise uses Python's “ThreadPoolExecutor” to run both “minimum” and “suitable” recommendation chains simultaneously. This significantly improves the system's speed, especially when multiple components are being recommended.

Code:

with `ThreadPoolExecutor()` as executor:

```
futures = {  
    executor.submit(recommend_hardware_with_llm, details, rec_type): rec_type  
    for rec_type in ["minimum", "suitable"]  
}
```

7.1.4 Data Storage and API Integration

Each recommendation is saved to the database through the “HardwareRecommendation” model and associated with the user’s query history. The API endpoint “Hardware_recommendation” handles this entire process:

- Retrieves the query and extracted details
- Calls the recommendation function for both types
- Stores the results in the database
- Returns a structured JSON response to the frontend

This modular structure allows the system to be easily extended or scaled in the future.

7.2 Web Scraping System

To ensure that SpecWise provides up-to-date hardware recommendations, we built a custom web scraper that extracts product data from e-commerce platforms, specifically BestBuy in this case. The scraper gathers real-time information including product names, prices, ratings, reviews, and product URLs. This information helps SpecWise match AI-based recommendations with components that are actually available in the market.

7.2.1 Scraping Strategy

The scraper is built using **Selenium** for browser automation and **BeautifulSoup-like parsing** via CSS selectors. It navigates through paginated search results across multiple hardware categories, simulating human-like behavior by introducing randomized delays and scroll actions to avoid detection.

7.2.2 Categories Covered

We scrape 7 essential PC hardware categories:

- CPUs
- GPUs
- RAM
- HDDs
- SSDs
- Motherboards
- Power Supplies

Each category has a unique search URL defined in a dictionary. The script loops through each and scrapes a configurable number of pages.

7.2.3 Key Features of the Scraper

1. Dynamic Page Handling:

The scraper waits for product elements to load using Selenium's "WebDriverWait", ensuring compatibility with dynamically rendered pages.

2. Real-Time Data Extraction:

For each product, we extract:

- Name
- Price
- Product link
- Rating
- Number of reviews
- Timestamp

Code:

```
name_elem = product.find_element(By.CSS_SELECTOR, ".sku-title a")  
price_elem = product.find_element(By.CSS_SELECTOR, ".priceView-customer-  
price span")  
name = name_elem.text  
link = name_elem.get_attribute("href")  
price = price_elem.text  
rating, reviews = extract_rating_and_reviews(rating_text)
```

7.2.3 Change Detection and Deduplication:

Using a “deduplicate_and_log()” function, the script compares new data with existing records. If any changes in price, rating, or reviews are found, they are logged in a text file. New entries are appended to a local JSON file, which acts as a lightweight product database.

7.2.4 Automated Logging:

All changes including additions and updates are saved in a log file with timestamps, making it easy to track how prices and reviews change over time.

Example log file:

```
[2025-04-06 01:50:59] Updated: Intel - Core i7-14700K 14th Gen 20-Core 28-Thread  
- 4.3GHz (5.6GHz Turbo) Socket LGA 1700 Unlocked Desktop Processor - Multi  
Changes: reviews (Old: 168, New: 198)  
  
[2025-04-06 01:50:59] Added: AMD - Ryzen 7 9800X3D 8-Core - 16-Thread 4.7  
GHz (5.2 GHz Max Boost) Socket AM5 Unlocked Desktop Processor - Silver  
  
[2025-04-06 01:51:00] Updated: AMD - Ryzen 9 7900X 12-core - 24-Thread 4.7  
GHz (5.6 GHz Max Boost) Socket AM5 Desktop Processor - Silver  
Changes: price (Old: $369.00, New: $374.00), reviews (Old: 94, New: 130)
```

7.2.5 Flexible File Output:

Scraped data is saved in a human-readable, indented JSON file for use by the backend. This also allows us to keep archives as JSON files for future use.

7.2.6 Execution Workflow

The script is run as a standalone Python file. When executed, it:

1. Launches a Chrome browser using “webdriver_manager”.
2. Iterates through the hardware categories.
3. Scrapes up to the pages given per category.
4. Updates and saves the data in JSON format.

Code:

```
if __name__ == "__main__":  
  
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))  
  
    try:  
  
        for cat, url_path in CATEGORIES.items():  
  
            scrape_bestbuy_category(driver, url_path, cat, JSON_path, log_path)  
  
    finally:  
  
        driver.quit()
```

7.3 *Technical Interface*

7.3.1 Fuzzy Search Hardware API Interface:

This API helps users search for hardware by name or type using fuzzy matching, making it easier to find products even with spelling variations or partial input.

Input:

- hardware_name (required): The name or partial name of the hardware to search for.
- hardware_type (required): The type of hardware (e.g., CPU, GPU).
- store_id (optional): Filter by store.

Output:

A list of products matching the fuzzy search query, with details like name, category, and price.

Code:

```
class FuzzySearchHardwareAPIView(APIView):  
    def get(self, request):  
        hardware_name = request.query_params.get('hardware_name', '')  
        hardware_type = request.query_params.get('hardware_type', '')  
        store_id = request.query_params.get('store_id', None)  
  
        # Filter products based on category and optional store_id  
        products = Product.objects.filter(category_icontains=hardware_type)  
        if store_id:  
            products = products.filter(store_id=store_id)  
        product_names = [product.name for product in products]  
  
        # Loop for fuzzy matching the hardware name  
        best_matches = process.extract(hardware_name, product_names,  
                                       scorer=fuzz.token_sort_ratio, limit=3)  
        matched_products = []
```

```

for match_name, score, _ in best_matches:
    try:
        product = products.get(name=match_name)
        matched_products.append(ProductSerializer(product).data)

    except Product.DoesNotExist:
        continue

    return Response(matched_products, status=status.HTTP_200_OK)

```

7.3.2 Sync Products interface

This API synchronizes the scraped product data into the Django database. The key part is handling the update or creation of product entries based on the link because the link is unique for every product.

Input:

A JSON object or a list of JSON objects containing product details like name, category, price, rating, and link.

Output:

A JSON response indicating the result of the sync operation, including how many products were created or updated into the database.

Code:

```

class SyncProductsAPIView(APIView):
    def post(self, request):
        products = request.data if isinstance(request.data, list) else [request.data]
        synced = []
        created_count = 0
        updated_count = 0

```

for item in products:

```
link = item.get("link")
if not link:
    continue # Skip if no link
# Update or create product based on the link
# (logic of updating/creating product follows)
```

7.3.3 Search Hardware interface

This API classifies a given search query and returns the predicted hardware category using a trained machine learning model.

Input:

query (required): A string containing the user's search input (e.g., "gaming mouse").

Output:

A JSON object with the predicted category of the hardware.

Code:

```
class ClassifyQueryAPIView(APIView):
    def post(self, request):
        query = request.data.get('query', "")
        if not query:
            return Response({ "error": "Query cannot be empty." }, status=400)

        # Preprocess and predict category
        cleaned = clean_text(query)
        vector = tfidf_vectorizer.transform([cleaned])
        predicted = model.predict(vector)[0]
```

```
return Response({ "category": predicted }, status=200)
```

7.4 Architecture Diagrams

Following diagrams provide a visual overview of the architecture, component interaction and deployment of this project.

7.4.1 Operational Flow diagram

A. LLM recommendation

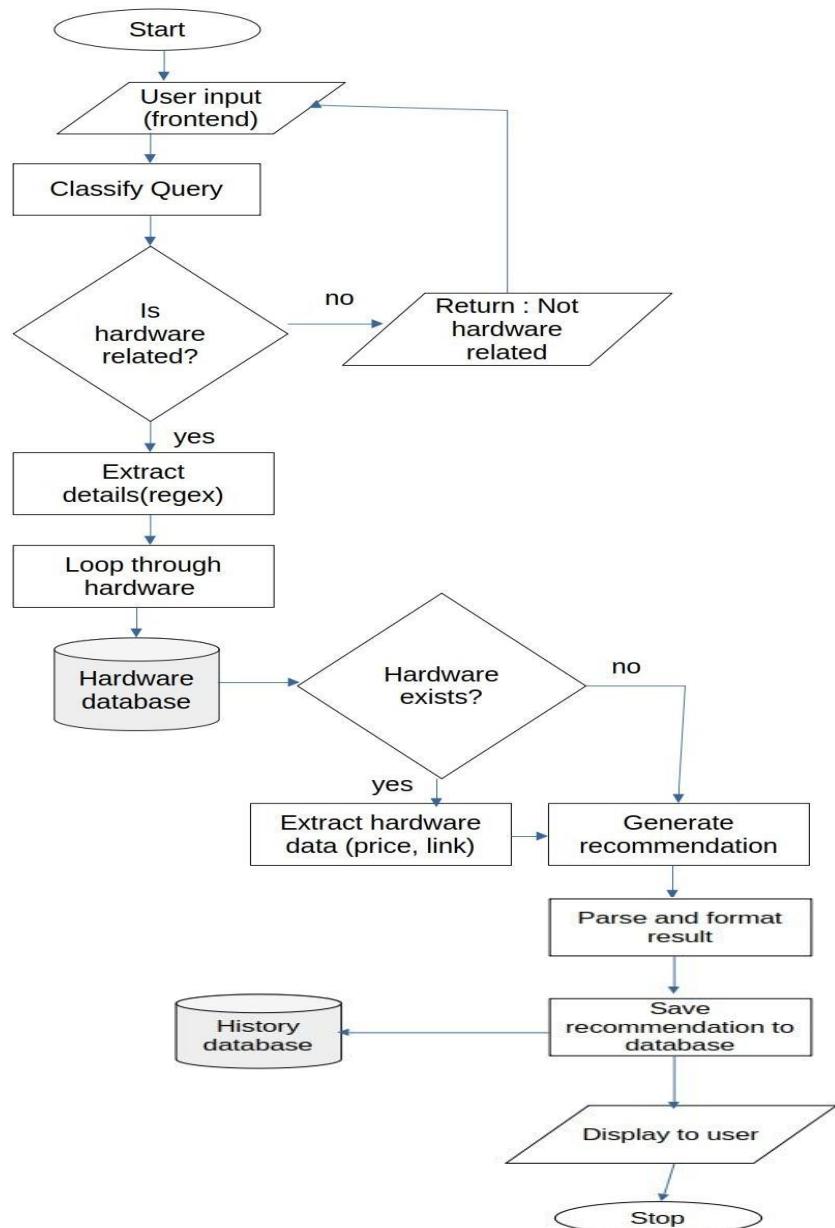


Figure 30: Specwise Operational Flow Diagram

B. Web scraping flow

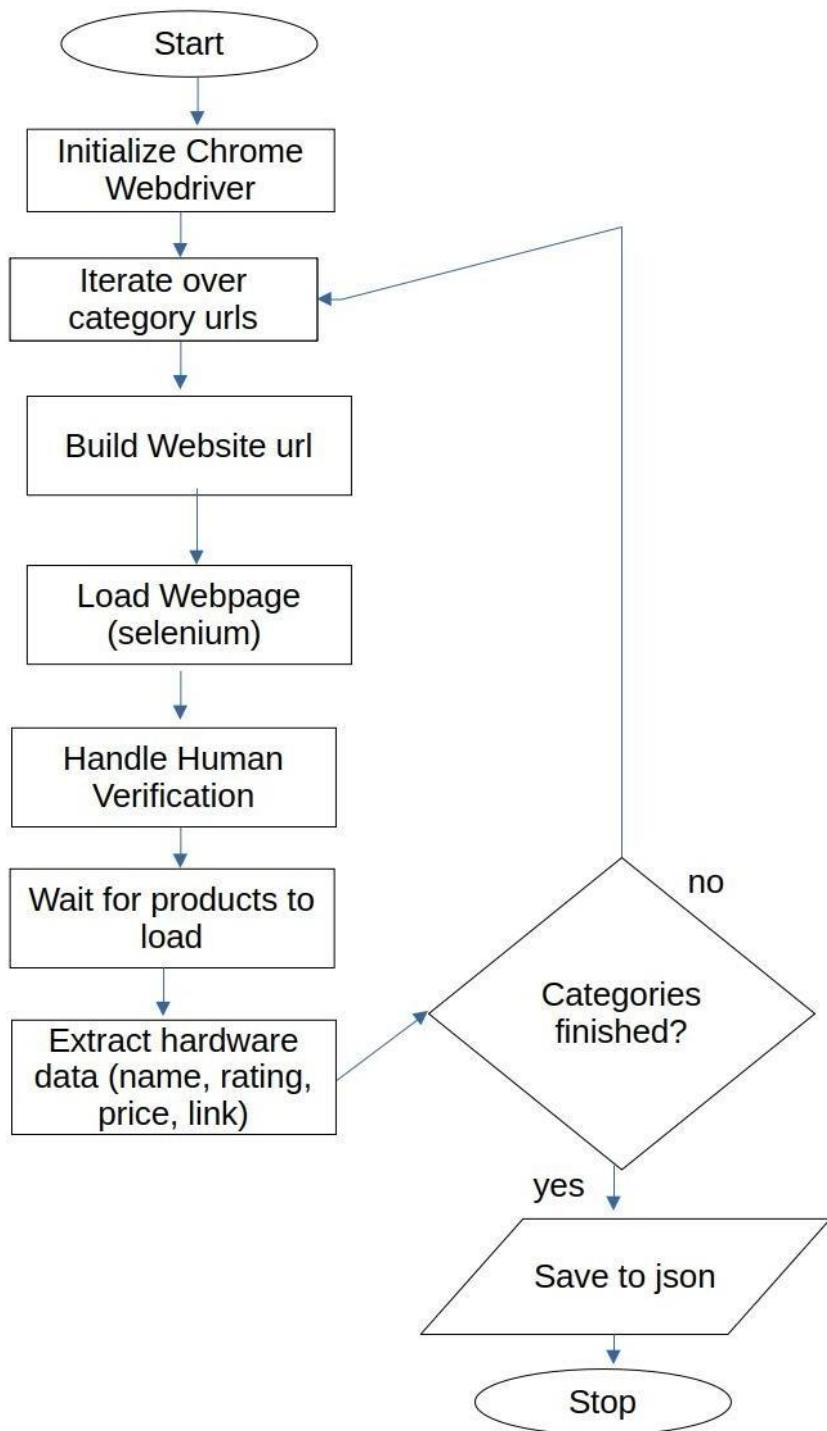


Figure 31: Web scraping Operational Diagram

7.4.2 Component Diagram

The component diagram is used to show the main modules or parts of the system and how they interact with each other.

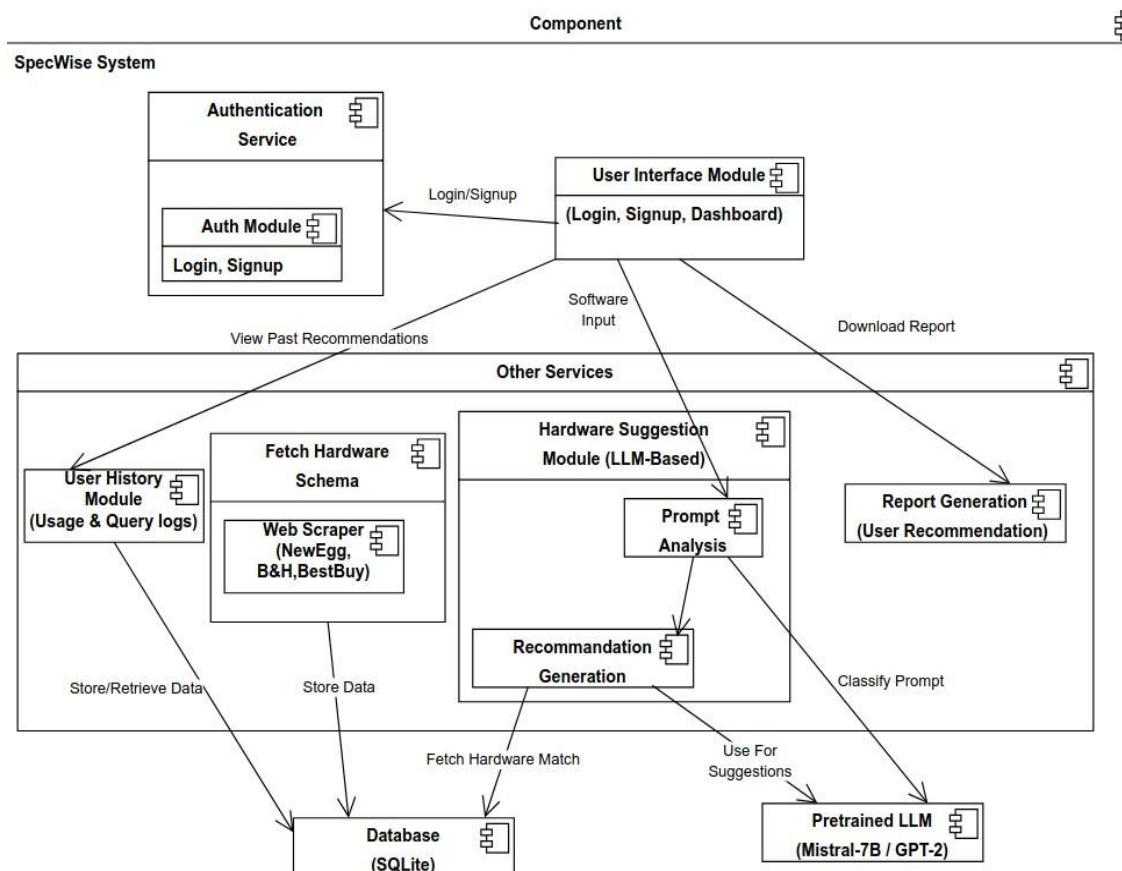


Figure 32: Component Diagram

7.4.3 Deployment Diagram

The deployment diagram illustrates how the system is physically deployed showing where each component lives and runs.

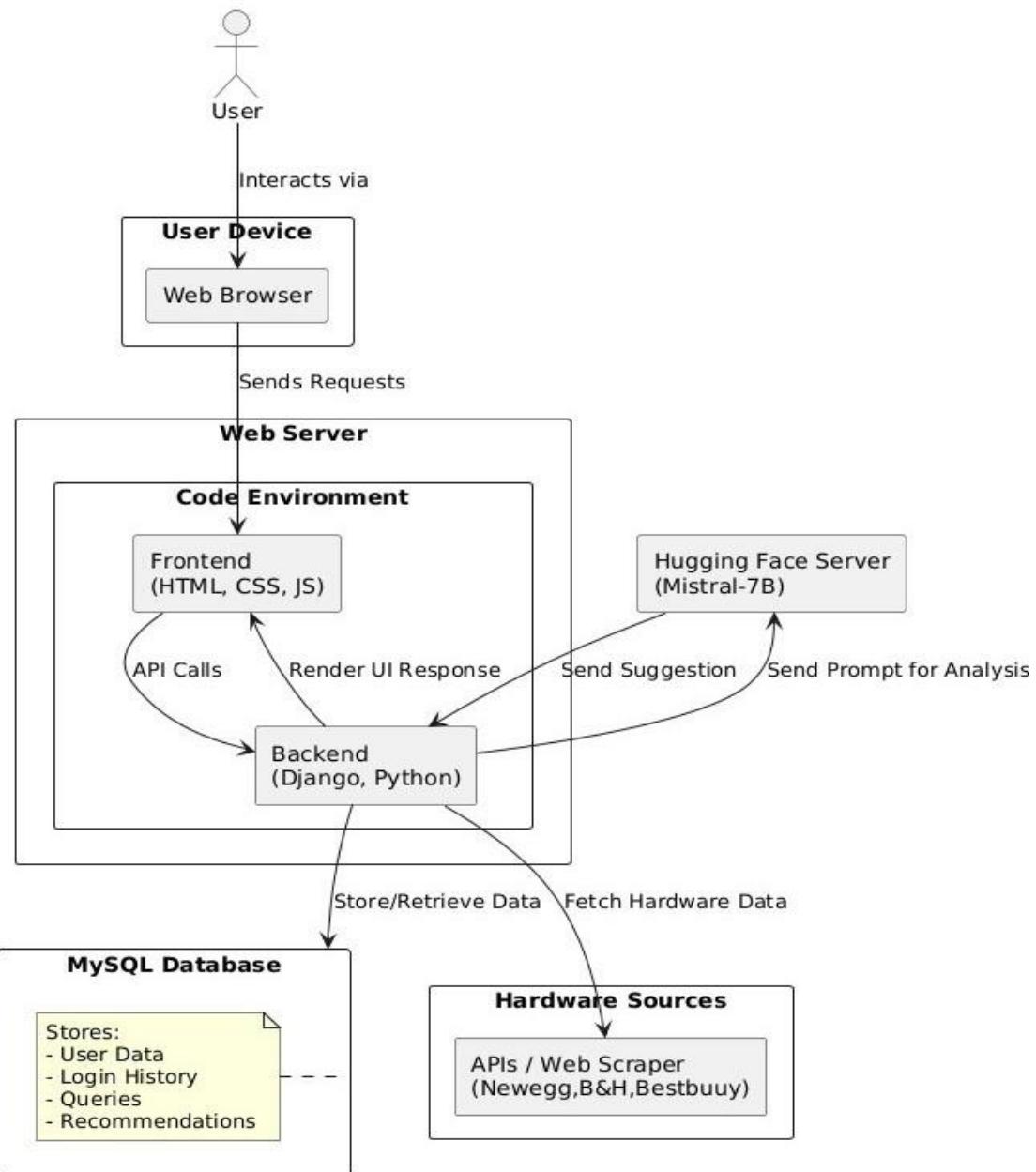


Figure 33: Deployment Diagram

8 Testing

8.1 White Box test cases

Id	Test case	Input	Expected Output	Actual Output	Status
WB-01	Sync Products – Missing Parameter	{ "name": "Intel - Core i5-14600K, "price": "\$259.99", "rating": , },	Rating missing, Continue without entering product	Failed to sync intel - Core i5-14600K - Status 500	Pass
WB-02	Sync Products – Valid inputs	Json containing products with valid inputs	Returns a confirmation message that all items are synced , status 200	{ "created": 0, "updated": 5941, "message": "Sync completed." }	Pass
WB-03	Fuzzy Search - Valid Input	hardware_name=Ryzen &hardware_type=CPU	JSON product list	Product list	pass
WB-04	Fuzzy Search - missing hardware name	hardware_type=CPU	All products with CPU types	{ "error": "Missing hardware_name or hardware_type" }	Fail

WB-05	Fuzzy Search - Hardware not found	hardware_name=abcd& hardware_type=GPU	Empty list	Product list	Fail
WB-06	Classify Query – Valid input	{ "question": "What graphics card is best for Blender rendering?", }	Determines that the question is hardware related	{ "question_id": 164, "is.hardware.related": "yes" }	Pass
WB-07	Classify Query – Out of credits	{ "question": "What graphics card is best for Blender rendering?", }	Determines that the question is hardware related	402 Client Error: Payment Required for url: https://router.huggingface.co/hf-inference/mistralai/Mistral-7B-Instruct-v0.3	Fail

Table 5: White Box Test Cases

8.2 Black Box Test cases

Id	Test case	Test Scenario	Expected Output	Actual Output	Status
BB-01	User wants to use the website without logging in	User inputs their query	Recommendations according to user's needs	Recommendations are displayed with their additional information	Pass
BB-02	Users want to view their history of recommendations	User is not logged in	User should not be able to view their recent history	User needs to be logged in in order to view history	Pass
BB-03	User wants to review a product	User enters a very short or a review that is not relevant	The review is not submitted because it barely provides any context	The review is submitted	Fail
BB-04	User wants a recommendation	User enters their input	The recommendation is given according to user;s needs	Their is a downtime or maintenance which blocks the usage of the model	Fail
BB-05	User is using the website on the go	User is using the website on mobile	The user expects a seamless experience	Some website features might break on different devices	Fail
BB-06	User wants a recommendation	User enters a query that is not hardware related	The system should give out an error	The system returns a non hardware query error and prompts the user to try again	Pass
BB-07	User creates enters a review	The Users enters nothing in both the fields	The system should prompt the user to enter something in the fields	The review is submitted	Fail
BB-08	The user enters the same query.	The user accidentally enters the same query twice and gets the same output	The system compares both queries and saves only one.	The queries show as two different queries in the review and history section	Fail

Table 6: Black Box Test Cases

9 Discussion

Throughout the SpecWise project, we embarked on a journey to solve a common problem: helping users, especially those who aren't tech-savvy, navigate the complex world of PC hardware selection. This chapter discusses the progression of the project, the methods we used, the experimental results we observed, and how these pieces collectively validate our design.

9.1 Project Progression and Methodology

We began by identifying the key challenge facing PC builders—a significant gap between the technical requirements of modern software and the typical consumer's ability to understand hardware specifications. To address this, we:

- **Conducted in-Depth Research:** We analyzed popular platforms like PCPartPicker, Logical Increments, BuildMyPC, and others to understand both their strengths and their shortcomings. This research provided valuable insights that shaped our requirements and design decisions.
- **Developed a Hybrid Recommendation System:** Our design integrates a pre-trained Large Language Model (Mistral-7B) to process user inputs and generate tailored hardware recommendations. In parallel, we implemented a web scraping mechanism to fetch real-time data from several e-commerce sites. This dual approach not only enhances the accuracy of our suggestions but also ensures the recommendations stay current.
- **Designed a User-Friendly Interface:** The front-end, built using React, was crafted to be intuitive, enabling users to input their software requirements with ease. Our backend, built with Django, handles data management and seamlessly integrates with the LLM and scraping modules.

9.2 Testing and Experimental Results

A significant part of the project's discussion revolves around testing—both to verify functionality and to evaluate overall performance:

- **White-Box Testing:** We conducted detailed white-box tests on critical code components such as the web scraper and LLM prompt system. The test cases focused on ensuring that:

- The scraper could handle multiple page traversals and correctly extract product data.
- The LLM prompt system accurately categorizes queries and produces viable hardware recommendations.
- **Black-Box Testing:** We also carried out black-box tests to simulate real-user interactions. For example, when a user inputs a scenario like “video editing with Adobe Premiere,” the system was tested for:
 - Speed and accuracy in returning relevant recommendations.
 - Usability of the recommendation interface

The testing results, summarized in our test plan (see Chapter 8), demonstrated that the system reliably meets its objectives. In cases where minor deviations were observed—such as delays when the scraper encountered dynamically loaded content—these were documented and noted as areas for future improvement.

9.3 Evaluation of the Design

From the experimental results and user tests, several key observations emerged:

- **Robustness of the Recommendation Engine:** The integration of the LLM with real-time data allowed SpecWise to produce hardware recommendations that not only fit users’ needs but also reflected current market prices and reviews.
- **Adaptability to Changing Web Environments:** One of the challenges identified was the dynamic nature of e-commerce website layouts, which sometimes affects the accuracy of the scraper. However, by using adaptive scraping techniques (such as human-like scrolling and deduplication mechanisms), the system has proven resilient under varied conditions.
- **Scalability and Future-Proofing:** The experimental phase highlighted the limitations of using SQLite in a production environment. This naturally leads into our recommendations for migrating to MongoDB as user data and system complexity grow. Additionally, fine-tuning our LLM with domain-specific data emerged as a promising step to further enhance recommendation precision.

9.4 Concluding Thoughts

The SpecWise project is a testament to how combining artificial intelligence with real-time data acquisition can simplify complex decision-making processes. Our systematic testing and experimental validation support the notion that our design not only works but also adds significant value to users seeking personalized PC hardware recommendations.

The comprehensive discussion of our methods backed by detailed experimental results from the testing chapter reinforces that while some uncertainties and challenges remain (mostly around adapting to dynamic e-commerce data and scaling the system), they are both manageable and part of a natural progression towards a more refined system.

In conclusion, SpecWise has built a solid foundation for an intelligent hardware recommendation service. With planned enhancements in scalability and further model optimization, we are confident that the system will continue to meet and exceed user expectations, proving its worth both in current applications and future expansions.

10 Conclusions

We can confidently say that our approach to simplifying PC hardware selection has proven its worth. By combining the strengths of AI-driven recommendations with real-time web data, we've built a system that makes choosing the right components very easy.

Our work shows that:

- **User Experience is Key:** With an intuitive design, users can easily enter their software needs and receive clear, tailored hardware recommendations.
- **Real-Time Data is a Game Changer:** The web scraping feature keeps the pricing and reviews up to date so that the suggestions remain relevant and practical.
- **Smart Recommendations:** The LLM, even in its pre-trained state, does a good job breaking down user queries and matching them with suitable hardware options.

That said, some uncertainties still need addressing:

- **Evolving Web Data:** E-commerce websites change their layouts frequently, which might require us to update or improve our scraping techniques over time. We're aware of this and have plans to keep our methods flexible.
- **Scaling Up:** The current use of SQLite is great for the initial phase, but we recognize that as our user base grows and data increases, switching to a more scalable solution like MongoDB will be important.
- **Model Fine-Tuning:** While the current pre-trained model works well, further customization using project-specific data could improve its precision. We're open to tweaking performance parameters and periodically retraining the model as new hardware trends emerge.

Overall, the design of SpecWise has demonstrated that our method can effectively address the challenges of PC hardware selection. We're excited about the possibilities for refining and expanding the system, and we believe that with a few adjustments along the way, SpecWise is well equipped to evolve into a highly reliable and user-friendly solution. Furthermore, SpecWise emphasizes continuous improvement by incorporating user feedback and adapting to emerging hardware trends, ensuring that the platform remains relevant and valuable over time.

The integration of AI and real-time data not only streamlines the hardware selection process but also empowers users with confidence and clarity in their purchasing decisions.

Looking ahead, we plan to explore advanced features such as personalized recommendation tuning, integration with additional e-commerce platforms, and enhanced community-driven insights to further enrich the user experience.

In summary, SpecWise represents a significant step forward in making PC hardware selection accessible, efficient, and intelligent for users of all backgrounds.

11 Future work

While SpecWise is already capable of providing smart hardware recommendations, there's plenty of room for growth. Here are some ideas we're excited to explore going forward:

- **Price History Tracking and Visualization**

We plan to add a feature that not only shows current hardware prices but also tracks price trends over time. By displaying interactive charts, users can see historical price fluctuations, which may help them decide on the best time to make a purchase.

- **Real-Time Notifications**

Users will be able to receive alerts on price drops, new product releases, or changes in availability, keeping them informed without needing to constantly check the platform.

- **Recommendations Based on Existing Hardware**

In the near future, we'd like to offer tailored advice based on what users already own. Instead of suggesting a completely new build, SpecWise could analyze existing components to recommend only those upgrades that make the most sense helping users balance performance and budget effectively.

- **Migration to MongoDB**

As our user base grows, we expect the data complexity to increase. Moving from SQLite to MongoDB would give us more flexibility when handling the diverse types of data.

- **Collaboration Features**

We envision collaborative build planning where users can share their builds with friends or experts for feedback and advice.

- **Customized Model Training**

Although Mistral-7B already provides solid recommendations, we see an opportunity to refine the advice even further. By training the model with data specific to PC hardware performance and user preferences, we aim to make our recommendations even more accurate and personalized.

- **Integration with Retailer APIs**

To complement web scraping, we plan to integrate directly with retailer APIs where available. This will provide more reliable and faster access to product data, pricing, and availability.

- **Community Reviews and Peer Builds**

Adding a social dimension could significantly enrich the user experience. In future releases, we envision a platform section where users can share their custom builds, post reviews, and offer tips.

- **Cloud-Based Build Simulation**

Offering a cloud-based simulation environment where users can virtually test their builds for performance and compatibility before purchase.

- **Accessibility Enhancements**

Continuous improvements to ensure the platform is accessible to users with disabilities, including screen reader support and keyboard navigation.

- **Multi-Language Support**

To broaden accessibility, SpecWise will support multiple languages, making the platform usable by a global audience.

- **Mobile App Companion**

Finally, we're looking into developing a mobile version of SpecWise. A lightweight app would let users check recommendations, view pricing trends, or verify compatibility on the go.

12 Achievements

this project has definitely been a learning journey for us. Over the past few months, we've picked up new skills, solved real problems, and built something from scratch and that, in itself, feels like a big achievement.

What We've Learned

- We learned how to build a working web application where users can log in, register, submit recommendations, and even review what they receive. It might sound simple on paper, but making it all work together smoothly took a lot of trial and error.
- We got hands-on with Django for the backend and used REST APIs to connect everything. It helped us understand how real-world websites work behind the scenes.
- We also improved how we write cleaner code, test features properly (like black-box testing), and handle user input securely.

Milestones We Hit

- Set up the entire user system — registration, login, history tracking — from scratch.
- Built APIs for handling search and classification, with custom logic that actually gives useful outputs.
- Managed to connect the frontend with the backend, which was a challenge in itself.
- Made sure users can review what they get and track their past interactions — something that involved a lot of backend work.

What This Project Gave Us

- Confidence in using tools like Django and React/Vite.
- A better understanding of how to structure a full-stack project.
- Problem-solving skills, especially when things broke (which was often).
- A sense of ownership this is something we built ourselves.

13 Appendices

Appendix A serves as a supplementary resource, offering additional technical details, supporting information, and practical examples that reinforce the main content of this report. This section is designed to provide readers with deeper insights into the project's development process, including configuration steps, sample data formats, and extended code examples. By presenting this information in an organized manner, Appendix A ensures that both technical and non-technical readers can better understand the implementation and operation of the SpecWise system.

13.1 *Appendix A*

13.1.1 Introduction

The SpecWise project seeks to streamline the difficult process of choosing PC hardware. It discusses typical problems people run into when selecting the appropriate parts for particular software programs. Conventional solutions frequently offer broad suggestions that could not meet the user's particular needs, which can lead to subpar performance or needless expenses. SpecWise employs artificial intelligence, notably pre-trained Large Language Models (LLMs), to assess user input and give optimum hardware options. This solution integrates a comparison engine to efficiently assess hardware components and employs real-time data scraping from multiple e-commerce websites to guarantee current recommendations.

13.1.2 Background

It is challenging for users to stay up to date with the latest technology due to the PC hardware industry's rapid evolution, which frequently results in confusion and less-than-ideal choices. Current systems, such as BuildMyPC, PCPartPicker, and Logical Increments, concentrate on hardware compatibility but are unable to offer customized suggestions suited to particular software workloads.

SpecWise sets itself apart by integrating AI-driven decision-making with LLMs, which take into account the most recent performance benchmarks, pricing, and reviews from e-commerce platforms in addition to analyzing user queries. This approach allows consumers to receive real-time, exact advice, bridging the gap between technological complexity and usability for non-expert users.

13.2 *Appendix B*

Appendix B provides an overview of the key tools and technologies utilized in the SpecWise project. It lists the web browsers, frameworks, databases, APIs, libraries, programming languages, and pre-trained models that form the foundation of the system. Additionally, it includes a comparison table highlighting how SpecWise stands out among similar websites by offering AI-powered recommendations and real-time pricing, features that enhance user experience and decision-making.

13.2.1 Tools Used

- Web Browsers: Google Chrome, Firefox, Microsoft Edge
- Frameworks: Django (Python), React (JavaScript)
- Databases: SQLite
- APIs: Hugging Face Transformers API
- Libraries: Pandas, Scikit-learn, Selenium
- Programming Languages: Python, JavaScript, HTML, CSS
- Pre-trained LLM: Mistral-7B via Hugging Face
- Web Scrapers: Selenium

13.2.2 Similar Websites

Feature	PCPartPicker	Logical Increments	BuildMyPC	SpecWise
Hardware Compatibility	Yes	No	Yes	Yes
AI-Powered Recommendations	No	No	No	Yes
Real-Time Pricing	No	No	Yes	Yes

13.3 Appendix C

Appendix C showcases important code snippets that demonstrate the core functionalities of SpecWise's web scraping and AI classification components. The web scraping snippet illustrates how the system extracts product details like name, price, rating, and reviews from BestBuy's website using Selenium. The LLM prompt classification snippet highlights how the AI model classifies user queries to determine if they are requesting hardware recommendations, enabling precise and relevant responses.

13.3.1 Web scraping snippet:

```
def scrape_bestbuy_page(driver, url):
    driver.get(url)
    handle_language_selection(driver)

    try:
        WebDriverWait(driver, 10).until(
            EC.presence_of_all_elements_located((By.CSS_SELECTOR, ".sku-item"))
        )
    except:
        return []
```

```

items = []
for product in driver.find_elements(By.CSS_SELECTOR, ".sku-item"):
    try:
        name_elem = product.find_element(By.CSS_SELECTOR, ".sku-title a")
        price_elem = product.find_element(By.CSS_SELECTOR, ".priceView-
customer-price span")

        name = name_elem.text
        link = name_elem.get_attribute("href")
        price = price_elem.text

    try:
        rating_text = product.find_element(By.CSS_SELECTOR, ".visually-
hidden").text
        rating, reviews = extract_rating_and_reviews(rating_text)
    except:
        rating, reviews = None, None

    items.append({
        "name": name,
        "price": price,
        "rating": rating,
        "reviews": reviews,
        "link": link,
        "last_updated": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    })
except:
    continue
return items

```

13.3.2 LLM Prompt Classification Snippet

```
template = """
```

```
You are a PC hardware recommendation classifier.
```

```
Answer 'Yes' or 'No' if the question explicitly requests PC hardware  
recommendations.
```

```
Question: {question}
```

```
Answer:
```

```
"""
```

```
response = LLM.invoke(template.format(question="What GPU is best for gaming?"))
```

```
classification = response.strip().lower()
```

```
print(classification)
```

14 References

- [1] R. B. Thompson and B. F. Thompson, "Building the perfect PC," 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:770041>
- [2] Statista. "Personal computer (PC) market in the U.S." [Online]. Available: <https://www.statista.com/topics/3149/personal-computer-pc-market-us/>.
- [3] S. Leek and D. Kun, "Consumer confusion in the Chinese personal computer market," Journal of Product & Brand Management, vol. 15, pp. 184-193, 2006. [Online]. Available: <https://api.semanticscholar.org/CorpusID:167743013>
- [4] PCPartPicker. (2024). "PCPartPicker: Create a Custom PC Build" [Online]. Available: <https://pcpartpicker.com>
- [5] Logical Increments. (2024). "Logical Increments: PC Buying Guide" [Online]. Available: <https://www.logicalincrements.com>
- [6] BuildMyPC. (2024). "BuildMyPC: Custom PC Builder" [Online]. Available: <https://buildmypc.net>
- [7] UserBenchmark. (2024). "UserBenchmark: Benchmark Your PC" [Online]. Available: <https://www.userbenchmark.com>
- [8] ChooseMyPC. (2024). "ChooseMyPC: Automatic PC Part Picker" [Online]. Available: <https://choosemypc.net>

- [9] E. Zennaro, L. Servadei, K. Devarajegowda, and W. Ecker, "A Machine Learning Approach for Area Prediction of Hardware Designs from Abstract Specifications," 21st Euromicro Conference on Digital System Design (DSD), pp. 413-420, 2018.
[Online]. Available:
<https://api.semanticscholar.org/CorpusID:52983611>
- [10] A. De Filippo, A. Borghesi, A. Boscarino, and M. Milano, "HADA: An automated tool for hardware dimensioning of AI applications," Knowledge-Based Systems, vol. 251, p. 109199, 2022. [Online]. Available:
<https://api.semanticscholar.org/CorpusID:251598344>
- [11] Z. Wang, S. Mao, W. Wu, and Y. Xia, "Assessing Phrase Break of ESL Speech with Pre-trained Language Models and Large Language Models," ArXiv, vol. abs/2306.04980, 2023. [Online]. Available:
<https://api.semanticscholar.org/CorpusID:253224099>
- [12] Hugging Face. "Mistral-7B Model Card." [Online]. Available:
<https://huggingface.co/mistralai/Mistral-7B>
- [13] I. S. H. Almaqbali, F. M. A. Al Khufairi, M. S. Khan, A. Z. Bhat, and I. Ahmed, "Web Scrapping: Data Extraction from Websites," Journal of Student Research, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:225792387>
- [14] R. Ur Rahman, D. Wadhwa, A. Bali, and D. S. Tomar, "The Emerging Threats of Web Scrapping to Web Applications Security and Their Defense Mechanism," 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:219464458>
- [15] The Selenium Project, "Selenium WebDriver Documentation." [Online]. Available: <https://www.selenium.dev/documentation/>

- [16] BeautifulSoup Documentation, [Online]. Available:
<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [17] Scrapy Documentation, [Online]. Available: <https://scrapy.org/>
- [18] Octoparse, "Web Scraping Tool." [Online]. Available:
<https://www.octoparse.com/>
- [19] P. Fettke and P. Loos, "Perspectives on Artificial Intelligence and its Applications in Recommender Systems," *Journal of Applied AI*, vol. 14, pp. 289- 310, 2007.
- [20] P. F. D. Ubois, *Python: Batteries Included*, 2007. [Online]. Available:
<https://api.semanticscholar.org/CorpusID:11271423>
- [21] S. Chen, S. Ahmed, K. Lal, and C. Deming, "Django Web Development Framework: Powering the Modern Web," *Asian Journal of Technology & Policy*, vol. 7, no. 3, 2022. [Online]. Available: <https://doi.org/10.18034/ajtp.v7i3.675>
- [22] S. bin Uzayr, *Mastering Django*, 2022. [Online]. Available:
<https://api.semanticscholar.org/CorpusID:252704721>
- [23] B. M. Carrasco Perez, J. R. Ramos, J. J. Hernandez Mora, M. C. Arenas, and M. J. Sanchez Hernandez, "Experience of using the Django ORM for Migration, Redesign and Implementation of a Database of MySQL Data to PostgreSQL," *International Journal of Science and Research (IJSR)*, vol. 10, no. 11, pp. 96–98, Nov. 2021. [Online]. Available: <https://www.ijsr.net/getabstract.php? paperid=SR211030134428>. doi: <https://www.doi.org/10.21275/SR211030134428>

- [24] “BlockchainSQLSecure: Integration of Blockchain to Strengthen Protection Against SQL Injections”, *BTSNUKPhM*, vol. 78, no. 1, pp. 160–168, Sep. 2024, doi: [10.17721/1812-5409.2024/1.29](https://doi.org/10.17721/1812-5409.2024/1.29).

Specwise

ORIGINALITY REPORT

12%	9%	3%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|---|--|-----|
| 1 | Submitted to Higher Education Commission Pakistan
Student Paper | 6% |
| 2 | arxiv.org
Internet Source | 1% |
| 3 | www.bestbuy.com
Internet Source | <1% |
| 4 | Submitted to Liverpool John Moores University
Student Paper | <1% |
| 5 | Yassine Attaoui, Mohamed Chentouf, Zine El Abidine Alaoui Ismaili, Aimad El Mourabit.
"Machine learning application for cell delay accuracy improvement at post-placement stage: A case study for combinational cells",
Integration, 2023
Publication | <1% |
| 6 | medium.com
Internet Source | <1% |
| 7 | Submitted to CSU, Dominguez Hills
Student Paper | <1% |
| 8 | Submitted to University of Wales central institutions
Student Paper | <1% |
| 9 | Submitted to Binus University International
Student Paper | |

		<1 %
10	tailor-network.eu Internet Source	<1 %
11	www.beei.org Internet Source	<1 %
12	Submitted to University of Derby Student Paper	<1 %
13	ar.scribd.com Internet Source	<1 %
14	iijecs.laescore.com Internet Source	<1 %
15	ir.ua.edu Internet Source	<1 %
16	Submitted to Flinders University Student Paper	<1 %
17	Submitted to Melbourne Institute of Technology Student Paper	<1 %
18	Submitted to Middle East College of Information Technology Student Paper	<1 %
19	Submitted to Colorado School of Mines Student Paper	<1 %
20	pure.royalholloway.ac.uk Internet Source	<1 %
21	laganvalleydup.co.uk Internet Source	<1 %
22	Tibor Gregorics. "Live & Learn", Teaching Mathematics and Computer Science, 2015	<1 %

Publication

23	Submitted to King's College Student Paper	<1 %
24	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
25	Submitted to University of Greenwich Student Paper	<1 %
26	Submitted to Asia Pacific Institute of Information Technology Student Paper	<1 %
27	Submitted to CTI Education Group Student Paper	<1 %
28	Submitted to Jain University Student Paper	<1 %
29	Rabi Jay. "Generative AI Apps with LangChain and Python", Springer Science and Business Media LLC, 2024 Publication	<1 %
30	Submitted to University of Bradford Student Paper	<1 %
31	paper.researchbib.com Internet Source	<1 %
32	research.birmingham.ac.uk Internet Source	<1 %
33	Submitted to Glasgow Caledonian University Student Paper	<1 %
34	export.arxiv.org Internet Source	<1 %
35	hdl.handle.net Internet Source	

		<1 %
36	kivanpolimis.com Internet Source	<1 %
37	patchwork.ozlabs.org Internet Source	<1 %
38	blog.enterprisedna.co Internet Source	<1 %
39	d-nb.info Internet Source	<1 %
40	repository.effatuniversity.edu.sa Internet Source	<1 %
41	www.coursehero.com Internet Source	<1 %
42	www.springerprofessional.de Internet Source	<1 %
43	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1 %
44	military.mypcls.org Internet Source	<1 %
45	www.asau.ru Internet Source	<1 %
46	www.emeraldinsight.com Internet Source	<1 %
47	Rizwan Ur Rahman, Danish Wadhwa, Aakash Bali, Deepak Singh Tomar. "chapter 53 The Emerging Threats of Web Scrapping to Web Applications Security and Their Defense Mechanism", IGI Global, 2020 Publication	<1 %

48

Zhou, Guanglin. "Navigating Distribution Shifts in ML through Causality and Foundation Models.", University of New South Wales (Australia)

<1 %

Publication

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off