

**Arab American University**

**Faculty of Engineering and Information Technology**

**Computer Systems Engineering**

**Software Engineering course**

**Student’s Names:**

**Mohammad Atteyah 202112724**

**Ibrahem Atatri 202111134**

**Baseem Al-jaghoob 202112412**

**Abstract**

AutoHub is a modern, complex web platform, exclusively designed and focused on serving automotive transaction needs for the Palestinian marketplace. Its absence has led to incredibly high levels of inefficiency for many car owners, dealers, and service providers who were constantly being exploited during transactions, losing so much precious time that could otherwise have been used productively.

The purpose of AutoHub is solely to address, alleviate, and streamline such gaps in service and transaction processes prevalent throughout by offering a friendly web interface with smart search capabilities, cross-compatible databases, data-importing capabilities, and a specially designed dealers' dashboard feature.

AutoHub is founded on a very robust system architecture based on a three-tier model for optimal performance and usability. It will have a comprehensive model consisting of a front-end module for user interface access-easy use for its patrons; a back-end for information processing, which would include processing and publishing data efficiently; and a safe database that allows efficient management both at transaction and listing levels.

Core competencies in the high-tech system include effective registration of its users, high volumes listed for cars and spare parts, AI-based smart search functions for maximum comfort of the user, secure payment processing capabilities that safeguard financial transactions, and review mechanisms allowing for feedback in light of experiences gained. On the technological tooling side, AutoHub relies on the latest technologies: React.js for frontend interfaces, Node.js for backend processing needs, PostgreSQL for robust database management, and AI-powered recommendation algorithms for optimized search efficiency with a high level of compatibility at the platform level. With a feasibility study supported with a thorough financial analysis, it is determined that not only is the project feasible but it holds high economic viability too. Several sources of revenue, including merchant subscriptions, collection of transactional fees, and advertisement streams, validate its conclusion. With strong practice in its management, compliance with strong security protocols, and expansion capabilities in the future, AutoHub is ready to transform the web marketplace for automobiles in Palestine. Along with its transformation, a significant boost in operational efficiency and a high level of improvement in overall experiences for all its participating users in such a vibrant marketplace will be realized.

Contents

[**1.** **Project Overview** 6](#_Toc189389866)

[**1.1.** **Title :** 6](#_Toc189389867)

[**1.2.** **Problem Statement :** 6](#_Toc189389868)

[**1.3.** **Objective:** 6](#_Toc189389869)

[**1.4.** **Project Scope :** 6](#_Toc189389870)

[**2.** **Requirements Gathering and Specification** 7](#_Toc189389871)

[**2.1.** **Stakeholder Identification** 7](#_Toc189389872)

[**2.2.** **Functional Requirements** 7](#_Toc189389873)

[**2.2.1.** **User Registration and Sign-in:** 7](#_Toc189389874)

[**2.2.2.** **Car Listings:** 7](#_Toc189389875)

[**2.2.4.** **Cross-Compatibility Feature:** 8](#_Toc189389876)

[**2.2.5.** **Import Service:** 8](#_Toc189389877)

[**2.2.6.** **Smart Search:** 8](#_Toc189389878)

[**2.2.7.** **Dealer Dashboard:** 8](#_Toc189389879)

[**2.2.8.** **Payment System** 8](#_Toc189389880)

[**2.2.9.** **User Reviews and Ratings:** 8](#_Toc189389881)

[**2.3.** **Non-Functional Requirements** 9](#_Toc189389882)

[**2.3.1.** **Performance:** 9](#_Toc189389883)

[**2.3.2.** **Scalability:** 9](#_Toc189389884)

[**2.3.3.** **Reliability:** 9](#_Toc189389885)

[**2.3.4.** **Security:** 9](#_Toc189389886)

[**2.3.5.** **Usability:** 9](#_Toc189389887)

[**2.3.6.** **Compatibility:** 9](#_Toc189389888)

[**2.3.7.** **Maintainability:** 9](#_Toc189389889)

[**2.4.** **Constraints** 10](#_Toc189389890)

[**2.4.1.** **Budget Constraints:** 10](#_Toc189389891)

[**2.4.2.** **Time Constraints:** 10](#_Toc189389892)

[**2.4.3.** **Technical Constraints:** 10](#_Toc189389893)

[**2.4.4.** **Legal Constraints:** 10](#_Toc189389894)

[**2.4.5.** **Resource Constraints:** 10](#_Toc189389895)

[**3.** **System Analysis** 10](#_Toc189389896)

[**3.1.** **Feasibility study :** 10](#_Toc189389897)

[**3.1.1.** **Executive Summary:** 10](#_Toc189389898)

[**3.1.2.** **Market Study:** 11](#_Toc189389899)

[**3.1.3.** **Financial and Time Study:** 12](#_Toc189389900)

[**3.1.4.** **Conclusion:** 16](#_Toc189389901)

[**3.2.** **Case Diagram :** 16](#_Toc189389902)

[**3.3.** **Data Flow Diagram :** 17](#_Toc189389903)

[**4.** **System Design** 18](#_Toc189389904)

[**4.1.** **System Architecture** 18](#_Toc189389905)

[**4.1.1.** **Overview:** 18](#_Toc189389906)

[**4.1.2.** **Client-Server Model:** 18](#_Toc189389907)

[**4.1.3.** **Database Structure:** 18](#_Toc189389908)

[**4.1.4.** **Third-Party Integrations:** 19](#_Toc189389909)

[**4.2.** **Sequence Diagrams** 19](#_Toc189389910)

[**4.2.1.** **User Searches for a Car** 19](#_Toc189389911)

[**4.2.2.** **User Places an Order** 19](#_Toc189389912)

[**4.2.3.** **Dealer Lists a Car** 20](#_Toc189389913)

[**4.3.** **Class Diagram:** 21](#_Toc189389914)

[**4.4.** **Prototype :** 22](#_Toc189389915)

[**5.** **AutoHub" Project Implementation Plan** 24](#_Toc189389916)

[**5.1.** **Selection of Tools and Programming language :-** 24](#_Toc189389917)

[**5.2.** **Programming Standards** 25](#_Toc189389918)

[**5.2.1.** **Naming Conventions** 25](#_Toc189389919)

[**5.2.2.** **Comments** 25](#_Toc189389920)

[**5.2.3.** **File Structure** 25](#_Toc189389921)

[**5.2.4.** **Indentation** 25](#_Toc189389922)

[**5.3.** **Version Control** 26](#_Toc189389923)

[**5.4.** **Software Modules** 26](#_Toc189389924)

[**5.4.1.** **Detailed Software Modules:** 26](#_Toc189389925)

[**5.4.2.** **Importance of Modularity** 27](#_Toc189389926)

[**5.5.** **Conclusion** 27](#_Toc189389927)

[**6.** **Deployment Strategy** 27](#_Toc189389928)

[**6.1.** **Environment Setup** 27](#_Toc189389929)

[**6.1.1.** **Server Setup:** 27](#_Toc189389930)

[**6.1.2.** **Database Configuration:** 28](#_Toc189389931)

[**6.1.3.** **Web Server Setup:** 28](#_Toc189389932)

[**6.1.4.** **Backend Setup:** 28](#_Toc189389933)

[**6.1.5.** **Frontend Setup:** 29](#_Toc189389934)

[**6.1.6.** **Third-Party Integrations:** 29](#_Toc189389935)

[**6.2.** **Installation Guide** 29](#_Toc189389936)

[**6.2.1.** **For Developers** 29](#_Toc189389937)

[**6.2.2.** **For End Users:** 30](#_Toc189389938)

[**6.3.** **Deployment Strategy** 30](#_Toc189389939)

[**6.3.1.** **Web-Based Deployment:** 30](#_Toc189389940)

[**6.3.2.** **Packaged Software (Optional)** 31](#_Toc189389941)

[**7.** **Project Management and Planning** 31](#_Toc189389942)

[**7.1.** **Work Breakdown Structure (WBS)** 31](#_Toc189389943)

[**7.2.** **Timeline:** 32](#_Toc189389944)

[**7.3.** Roles and Responsibilities 34](#_Toc189389945)

[**Conclusion** 35](#_Toc189389946)

# **Project Overview**

## **Title :**

"AutoHub: The Online Platform of Selling, Renting Cars and Sourcing of Spare Parts within the Palestinian Market"

## **Problem Statement :**

There is no established digital marketplace for buying, renting cars, and finding spare parts in Palestine. The current online platforms can do little other than just advertising cars. No decent service like cross-compatibility of all parts, import services, user-friendly interfaces, and so on exists. This makes owners of vehicles, repair shops, and house dealers suffer through inefficiencies, exploitations, and wastage of time.

## **Objective:**

AutoHub would like to create a full-fledged online hub that would make buying, renting cars, and supplying spare parts easier in the Palestinian market. The platform will include a user-friendly web interface, database cross-compatibility of car parts, and even importation service in case of part non-existence. The platform reduces exploitation, saves time while enhancing experiences for car owners, repair workshops, and other dealers.

## **Project Scope :**

**Included are:**

1. Car Sales and Rentals: View, compare, buy, and rent cars.
2. Spare Parts Database: Full database of original and modified spare parts with cross-compatibility features.
3. Import Service: Users can order parts not available locally through the platform.
4. User Accounts: Different account types: individual, dealer, and repair workshop.
5. Smart Search: AI-based search for cars and parts using user preference criteria.
6. Dealer Dashboard: A separate administration interface for the management of listings by the dealers.
7. Payment System: Subscription-based payment models both for dealers and shops.

**Out of scope:**

1. Physical Inventory: Car and spare parts physical inventory is not to be managed on this platform.
2. Manufacturing: Manufacturing and modification of the car parts would not be involved with this.
3. Delivery Services: The website will provide ordering, but the physical delivery shall not be carried out by it.
4. Legal Services: The platform will not render any legal services nor will it manage disputes between buyers and sellers.

**Primary Users:** Car owners, repair workshops, and dealers.

**Secondary Users:** Tenants who want to rent cars and those willing to import spare parts.

**Key Features:**

1. Cross-Compatibility Database: Users can view and compare different car brands.
2. AI-Powered Smart Search: Advanced search for cars and parts.
3. Import Service: Ordering parts from overseas suppliers.
4. Dealer Dashboard: Dealers can manage their listings and subscriptions in one place.
5. User Reviews and Ratings: The users can leave feedback about the products and the sellers.

# **Requirements Gathering and Specification**

## **Stakeholder Identification**

* **Stakeholders**:
* Project Sponsor: The organization or individual funding the project.
* Development Team: Developers, designers, and testers responsible for building the platform.
* Marketing Team: Responsible for promoting the platform and attracting users.
* Legal Advisors: Ensure compliance with local laws and regulations.
* Suppliers and Partners: International suppliers for spare parts and local repair workshops.
* **Users**:
* Car Owners: Individuals looking to buy, rent, or repair cars.
* Car Dealers: Businesses that sell or rent cars through the platform.
* Repair Workshops: Workshops that need spare parts for repairs.
* Spare Parts Sellers: Businesses that sell spare parts through the platform.
* Administrators: Platform managers who oversee operations and user accounts.

## **Functional Requirements**

### **User Registration and Sign-in:**

The user should be allowed to register on the platform, with further ability to sign in either by email or via social media services.  
Example: A car owner creates an account with his email address and a password.

### **Car Listings:**

Dealers can list cars for sale or rent, including details like price, model, and photos.  
Example: A dealer lists a 2020 Toyota Corolla for $15,000.

* + 1. **Spare Parts Database:**  
       Users can search for spare parts by car brand, model, or part number.  
       Example: A repair workshop searches for a brake pad compatible with a 2018 Skoda Octavia

### **Cross-Compatibility Feature:**

The system suggests compatible parts across different brands.  
Example: the platform would suggest that a Volkswagen brake pad is compatible with a Skoda Octavia.

### **Import Service:**

Customers can purchase a part not available locally, and the system will provide a quotation including import cost and taxes.  
Example: The customer orders an exotic engine part from Germany, and the system gives him the price including shipping and taxes for the import.

### **Smart Search:**

The users can search for cars or their parts based on filters like price, brand, model, and specification.  
Example: An application user searches for a family car within a $20,000 budget and gets SUV suggestions.

### **Dealer Dashboard:**

Able dealers to manage listings, sales statistics, and inventory.  
Example: A dealer logs in and looks at how many cars he has sold this month.

### **Payment System**

Users can pay subscription, buy cars, or spare parts through the platform.  
Example: A dealer subscribes annually at $300 to list their cars.

### **User Reviews and Ratings:**

Users can review and rate cars, parts, and sellers.  
Example: A car buyer gives a rating of his/her experience with a dealer as 5 stars.

## **Non-Functional Requirements**

### **Performance:**

The platform shall load the search results within 2 seconds.  
Example: A user searching for a car should get results in nearly real time.

### **Scalability:**

The platform shall support up to 10,000 concurrent users without degradation in performance.  
Example: The platform should be responsive during high traffic driven by promotional campaigns.

### **Reliability:**

The platform shall be available and up 99.9% of the time, with minimal downtime for maintenance.  
Example: Users should have access to the platform at any time without interruptions.

### **Security:**

All data of users and transactions should be encrypted using SSL/TLS.

Example: The credit card details of a user are securely stored and transferred.

### **Usability:**

The platform must be intuitive to use, with no requirement for technical skills to operate. Example: A new user can perform a car search without requiring any instructions.

### **Compatibility:**

The platform should work seamlessly on all major browsers- Chrome, Firefox, Safari- desktop, mobile, and tablet devices.  
Example: A user can access the platform on an iPhone using Safari.

### **Maintainability:**

The code is documented and adequately modularized; this would guarantee ease of any upgrade and quick bug fixes  
Example: Adding a new functionality should not compromise other functionalities.

## **Constraints**

### **Budget Constraints:**

Budget of 82,500$ that caters for developmental costs, hosting, and advertisement  
Example: The team would need to put a hold on feature requests as to be cost-effective

### **Time Constraints:**

The platform should be launched within 6 months.  
Example: Development and testing should be in a very tight schedule.

### **Technical Constraints:**

The platform must support the currently operating payment gateways in Palestine.  
Example: The payment system should be integrated with local banks and international payment processors.

### **Legal Constraints:**

The platform should respect the commercial laws of Palestine and regulations on import and export.  
Example: The import service should be compliant with customs and tax laws.

### **Resource Constraints:**

The development team is limited to 7 members, including developers, designers, and testers.  
Example: The team must work efficiently to deliver the project on time.

# **System Analysis**

## **Feasibility study :**

Feasibility Study for a Website Project of Selling and Renting Cars and Spare Parts in the Palestinian Market

### **Executive Summary:**

This project will involve developing an e-commerce website focused on the sale and rent of cars and selling original and modified spare parts, with a focus on the Palestinian market. This project involves a complete database about all types of car parts, whereby any user can compare between compatible parts between brands-for example, Skoda and Volkswagen-to avoid being exploited and wasting a lot of time. The website will also provide the service of importing the required parts from outside the Palestinian market.

### **Market Study:**

#### **Market Analysis:**

* **Market Size:** The Palestinian market is suffering due to the dire lack of trusted digital platforms able to satisfy the needs related to buying and renting cars and spare parts.
* **Target Audience**:
* Users seeking new or used cars either for rent or purchase
* Repair workshops, car dealers, and car owners require an integrated parts database
* Those willing to import the parts not available locally.

#### **Competitors:**

Existing platforms are limited to car advertisements only and do not provide comprehensive services or a complete database.

#### **Requirements**

* Platform that will easily and completely help the user.
* Updated constantly and with high accuracy, database.
* Technical as well as commercial support to users.

#### **Technical Study:**

#### **Technical features of the website:**

##### **Main Interface:**

* To display the cars for sale/rented.
* A list of original and modified car parts (for modifying appearance, Horsepower, and Mechanical).

##### **Database:**

* Includes all car types and their parts.
* A compatibility matching system (Cross-Compatibility).
* User profiles categorized depending on their registration-for example, car or parts dealer, car showroom with large stock, spare parts store with large stock, and regular users. The ability to save desired products, add them to the cart, and save user preferences to filter the search is provided, along with the ability to leave a review and comments on the products. In the future, an online chat between the sellers and buyers could be developed.
* Payment mechanism for the users based on their category:
* Showrooms: Annual subscription mechanism based on the number of cars displayed every month at a very nominal price.
* Online dealers: The ability to pay for each listing or monthly/annually based on the number of listings, whether for car sales, rentals, or spare parts.
* Large spare parts stores: Similar to the showroom payment model.

##### **Smart Search:**

* It helps users to quickly search for any part by a car or a manufacturer. The interface will present the user with all parts in an easy manner, showing all specific designs and codes.
* ntelligent car search would cover not only name, type, brand, and model but also specifications, prices, and even general preferences-for instance, family, sports, or youth cars, the capacity of the engine, and so on. For example, when the user needs a family car for four members with the requirement of a weekend trip, the search will suggest a high loading capacity SUV, with alternative suggestions, keeping in mind the budget and the needs, incorporating AI.

##### **Order System:**

Allows users to easily order parts from outside the Palestinian market by selecting the product from the displayed car parts and ordering through the platform with a prepayment payment system, including tax and import cost calculation.

##### **Dealer Dashboard:**

Special interface for dealers and spare parts shops to add their products.

##### **Required Technical Resources:**

o Web development team (Full-Stack Developers).

o UX/UI Designer.

o Strong servers to host the website and database.

o AI specialist.

o e-security system to protect data and transactions.

This is a general website that targets the overall market of Palestine. The changes will be made according to the suggestions of the users. Furthermore, this website is not for any personal or commercial use. The website will be developed gradually as, at the time of development, all the needs of the user cannot be precisely identified.

### **Financial and Time Study:**

#### **Design Cost**

##### **Timeline with Distribution of Salary (Average salary = $1,500)**

| **Stage** | **Required Roles** | **Number of People** | **Duration (in months)** | **Total Salaries (USD)** |
| --- | --- | --- | --- | --- |
| Analysis and Planning | Project Manager | 1 | 0.5 | 750 |
| UI/UX Design | UI/UX Designer | 1 | 1.5 | 2250 |
| Backend Development | Backend Developer + Database Engineer | 2 | 3 | 9000 |
| Frontend Development | Frontend Developer | 1 | 2.5 | 3750 |
| QA Testing | QA Tester | 1 | 1.5 | 2250 |
| Content Creation | Content Writer + SEO Specialist | 2 | 1.5 | 4500 |
| E-commerce Integration | Payment Gateway Developer | 1 | 1 | 1500 |
| Launch and Review | Full Team (7 people) | 7 | 0.5 | 5250 |
| **Total Salary Cost** |  |  |  | **29,250 USD** |

##### **Additional Costs**

| Item | Cost (USD) | Details |
| --- | --- | --- |
| Design Tools | 500 | Subscription for design software like Adobe XD or Figma |
| Hosting and Servers | 1200 | High-performance hosting for one year |
| Software Development Tools | 800 | GitHub subscriptions, development environments, and other tools |
| Purchase of Large Database | 10,000 - 50,000 | Depending on the size and type of required data |
| Other Costs | 750 | Includes domain purchase and advertisements |
| **Total Additional Costs** | 13,250 - 53,250 |  |

##### **Detailed Financial Timeline (Monthly)**

| **Month** | **Basic Expenses (Salaries)** | **Additional Costs (Minimum)** | **Total Monthly (USD)** |
| --- | --- | --- | --- |
| First Month | 3750 | 1500 | 5250 |
| Second Month | 6000 | 1000 | 7000 |
| Third Month | 7500 | 10,500 (Including Database) | 18,000 |
| Fourth Month | 5250 | 250 | 5500 |
| Fifth Month | 3000 | 0 | 3000 |
| Sixth Month | 3750 | 0 | 3750 |

**Total Costs:**

• Minimum Total Costs: $29,250 Salaries + $13,250 Additional = $42,500 USD

• Maximum Total Costs: $29,250 - Salaries and $53,250 - Additional = $82,500 USD. These costs are dependent on the size and details of the database purchased.

#### **Running Cost**

|  |  |  |
| --- | --- | --- |
| Item | Monthly Cost (USD) | Annual Cost (USD) |
| Hosting and Maintenance | 500 | 6,000 |
| Security and Updates | 300 | 3,600 |
| Marketing | 1,000 | 12,000 |
| Customer Support | 1,500 | 18,000 |
| **Total Running Costs** | **3,300** | **39,600** |

#### **Revenue Streams:**

##### **Revenue from Merchant Subscriptions for Displaying Cars and Spare Parts**

| **Item** | **Monthly Cost (USD)** | **Annual Cost (USD)** | **Details** |
| --- | --- | --- | --- |
| Exhibitor Subscriptions (200 Exhibitors) | 1,666 - 5,000 | 40,000 - 120,000 | Based on annual subscription ranging from $100-300 per shop or exhibitor |
| Spare Parts Shop Subscriptions (200 Shops) | 1,666 - 5,000 | 40,000 - 120,000 | Same subscription but for spare parts shops |

Total Monthly Revenue from Subscription: From $3,332 to $10,000 per month.

Total Annual Revenue from Subscription: From $80,000 to $240,000 per year.

##### **Revenue from Small Merchant Subscription for Display of Cars and Spare Parts:**

| **Item** | **Monthly Cost (USD)** | **Annual Cost (USD)** | **Details** |
| --- | --- | --- | --- |
| Small Merchant Subscriptions (1,000 Posts Monthly) | 1,000 Posts × ($0.1 to $1 per Post) = 100 - 1,000 | 1,200 - 12,000 | Based on the number of posts monthly and the number of merchants displaying their products |

Total Monthly Revenue: From $100 to $1,000.

Total Annual Revenue: From $1,200 to $12,000.

##### **Revenue from Sales Commissions and Orders**

| **Item** | **Monthly Cost (USD)** | **Annual Cost (USD)** | **Details** |
| --- | --- | --- | --- |
| Sales Commissions and Orders (1,000 Transactions Monthly × $3 per Order) | 3,000 per month | 36,000 per year | Based on 1,000 sales transactions per month with an average commission of $3 per transaction |

Total Monthly Revenue: $3,000

Total Annual Revenue: $36,000

##### **Revenue from Attracting Large Companies for Ads**

| **Item** | **Monthly Cost (USD)** | **Annual Cost (USD)** | **Details** |
| --- | --- | --- | --- |
| Ads for Large Companies (Local and Global) | 1,000 - 5,000 per month | 12,000 - 60,000 per year | Based on ads from local and global large companies |

Total Monthly Income: Between $1,000 and $5,000

Total Annual Income: Between $12,000 and $60,000

#### **Total Income:**

##### **Total Monthly Income:**

| **Item** | **Monthly Cost (USD)** | **Details** |
| --- | --- | --- |
| Exhibitor and Shop Subscriptions | From $3,332 to $10,000 | Based on the number of subscribing exhibitors and shops |
| Small Merchant Subscriptions | From $100 to $1,000 | Based on the number of posts monthly |
| Sales Commissions and Orders | $3,000 | Based on 1,000 sales transactions monthly |
| Ads from Companies | From $1,000 to $5,000 | Based on ads from large companies |

Total Monthly Income: Between $7,432 and $19,000

##### **Total Annual Income:**

| **Item** | **Annual Cost (USD)** | **Details** |
| --- | --- | --- |
| Exhibitor and Shop Subscriptions | From $80,000 to $240,000 | Based on annual subscriptions for exhibitors and shops |
| Small Merchant Subscriptions | From $1,200 to $12,000 | Based on small merchant subscriptions for posts |
| Sales Commissions and Orders | $36,000 | Based on commission from sales |
| Ads from Companies | From $12,000 to $60,000 | Based on ads shown on the site |

Total Annual Revenue: $129,200-$348,000

### **Conclusion:**

* **Profitability:** The project looks highly profitable with substantial monthly and yearly revenues from various income sources.
* **Cost Recovery:** In light of the initial project costs related to development, marketing, and hosting, the project looks capable of recovering those costs within its first year, especially considering forecasted income generated by advertisements and commissions on sales.

**Legal Study:**

* + Registration of the project as a company in the Palestinian market
  + Commercial laws observed
  + Import and export permits obtained

**Contingent Dangers:**

1. Competition: New entrants with competing offerings threaten market share.

* Plan: Differentiate services by constantly updating the website.

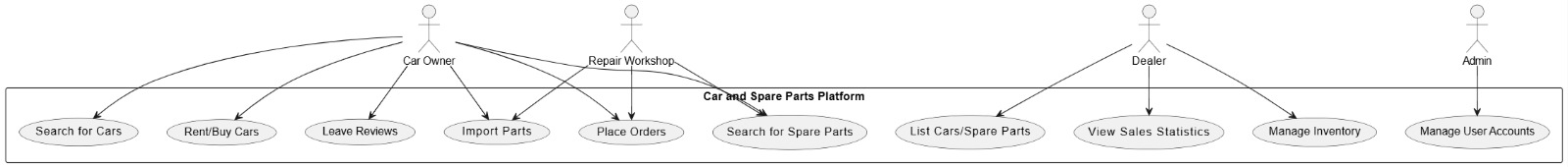
1. Import Delays:

* Strategy: Have a strong network of suppliers across the world to ensure delivery as soon as possible.

**Recommendations:**

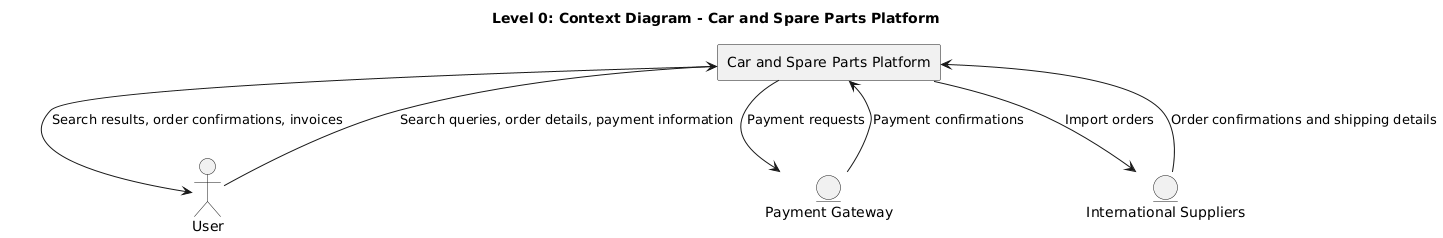
* Digital marketing will be done primarily to allure customers by putting a dedicated social media marketing team.
* Local tie-ups with repair workshops and spare parts shops.
* Special promotions and discounts for new users.
* Mee the basic requirements of the users, be updated and introduce a mobile app.

## **Case Diagram :**

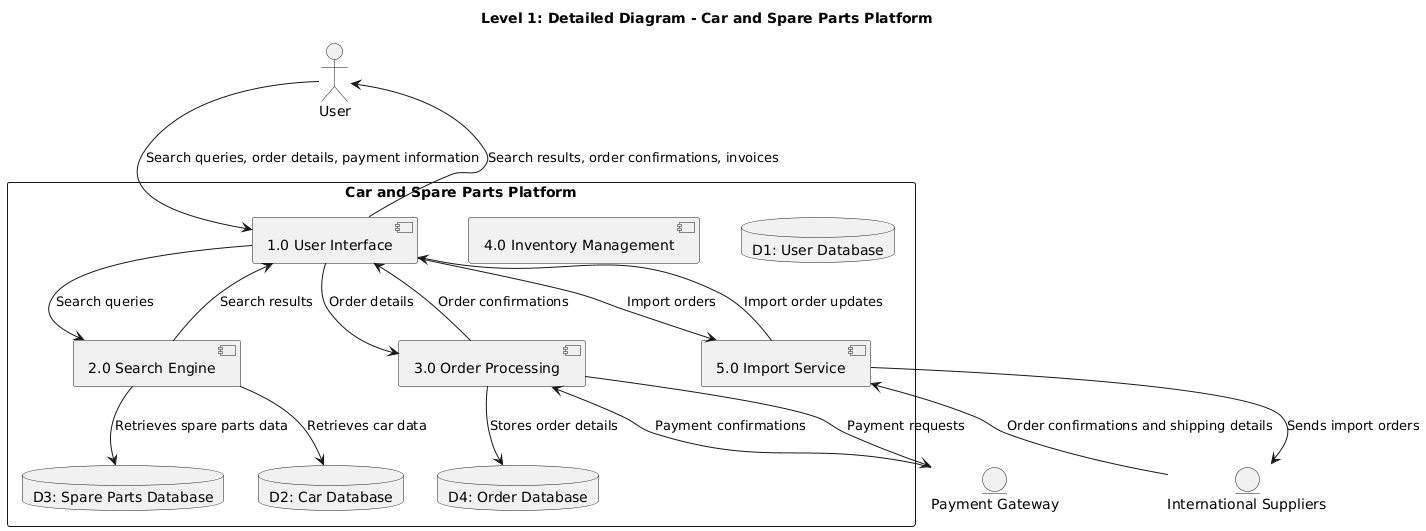


## **Data Flow Diagram :**

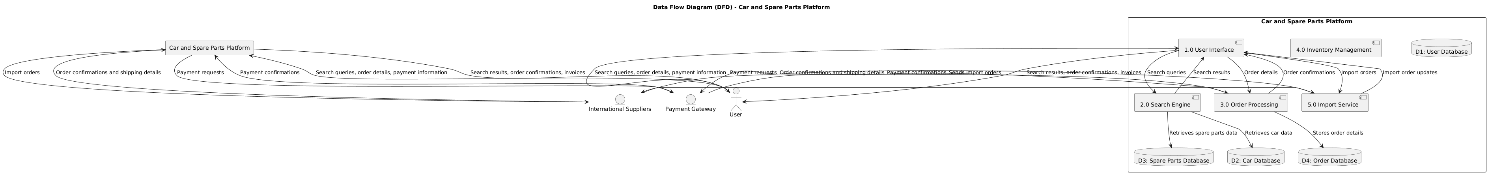
DFD level0:



DFD level1:



DFD level 0,1:



# **System Design**

## **System Architecture**

### **Overview:**

The system follows a three-tier architecture consisting of:

1. **Frontend (Presentation Layer):**

Handles the user interface (UI) and user experience (UX).

Technologies: HTML, CSS, JavaScript, React.js (or Angular/Vue.js).

1. **Backend (Application Layer):**

Handles business logic, data processing, and communication with the database.

Technologies: Node.js, Express.js (or Django, Spring Boot).

1. **Database (Data Layer):**

Stores all data related to users, cars, spare parts, orders, etc.

Technologies: MySQL, PostgreSQL, or MongoDB.

### **Client-Server Model:**

* **Client:** The frontend (user interface) runs on the user’s device (e.g., browser or mobile app).
* **Server:**The backend runs on a remote server and handles requests from the client, processes data, and interacts with the database.
* **Communication:** The client and server communicate via HTTP/HTTPS using RESTful APIs or GraphQL.

### **Database Structure:**

**Tables:**

1. User Table:

Fields: user\_id, name, email, password, role (e.g., car owner, dealer, admin).

1. Car Table:

Fields: car\_id, brand, model, year, price, availability, dealer\_id.

1. Spare Parts Table:

Fields: part\_id, part\_name, compatibility, price, stock, seller\_id.

1. Order Table:

Fields: order\_id, user\_id, product\_id, quantity, total\_price, status.

1. Dealer Table:

Fields: dealer\_id, business\_name, contact\_info, subscription\_status.

### **Third-Party Integrations:**

1. **Payment Gateway:**

Integrates with local and international payment processors (e.g., PayPal, Stripe, or local banks).

Handles payment processing for car purchases, rentals, and spare parts orders.

1. **AI/ML Services:**

Uses AI libraries (e.g., TensorFlow, Scikit-learn) for the smart search feature.

1. **Shipping and Logistics:**

Integrates with international suppliers and shipping companies for import orders.

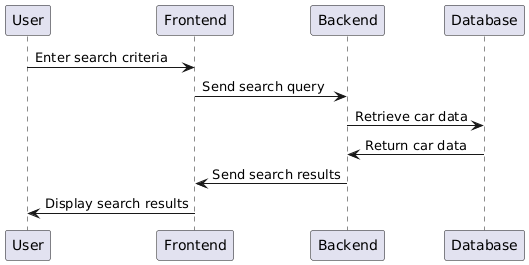
1. **Email/SMS Services:**

Sends notifications (e.g., order confirmations, payment receipts) via email or SMS.

## **Sequence Diagrams**

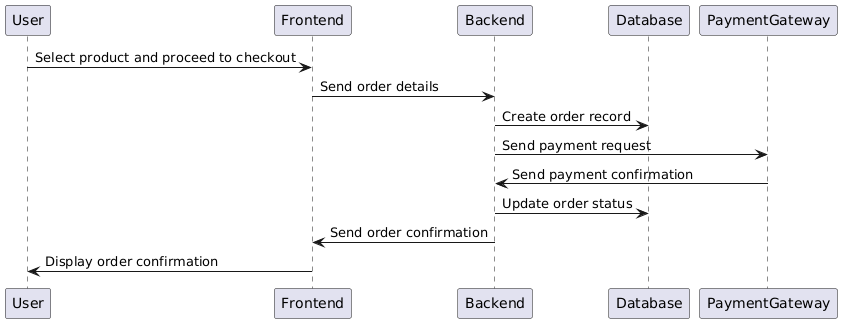
### **User Searches for a Car**

* Actors: User, Frontend, Backend, Database.
* Steps:
* User enters search criteria (e.g., brand, model, price range) on the frontend.
* Frontend sends the search query to the backend via an API call.
* Backend processes the query and retrieves matching car data from the database.
* Backend sends the search results back to the frontend.
* Frontend displays the results to the user.
* **Sequence Diagram for User Searches for a Car:**



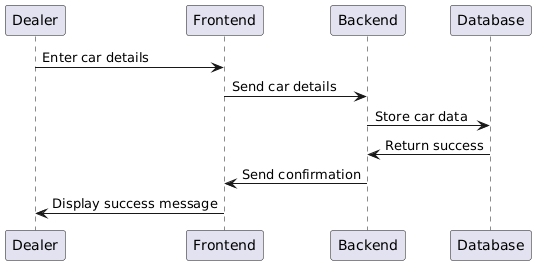
### **User Places an Order**

* Actors: User, Frontend, Backend, Database, Payment Gateway.
* Steps:
* User selects a car or spare part and proceeds to checkout.
* Frontend sends the order details to the backend.
* Backend creates an order record in the database.
* Backend sends a payment request to the Payment Gateway.
* Payment Gateway processes the payment and sends a confirmation to the backend.
* Backend updates the order status in the database.
* Backend sends an order confirmation to the frontend.
* Frontend displays the confirmation to the user.
* **Sequence Diagram for User Places an Order:**

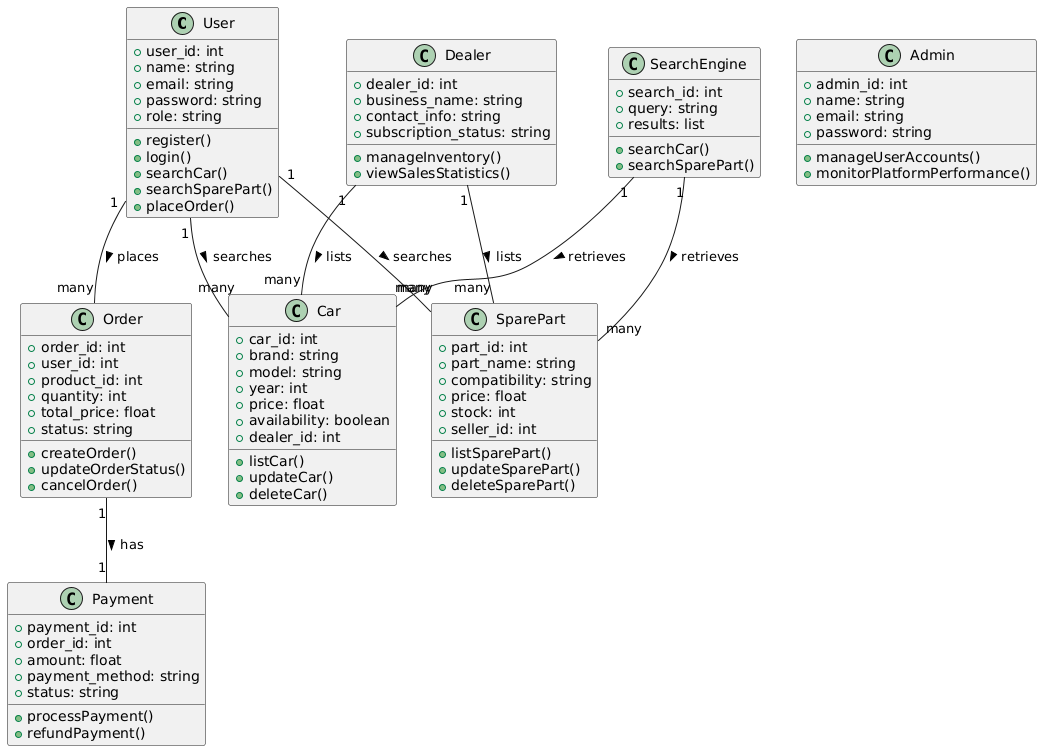
****

### **Dealer Lists a Car**

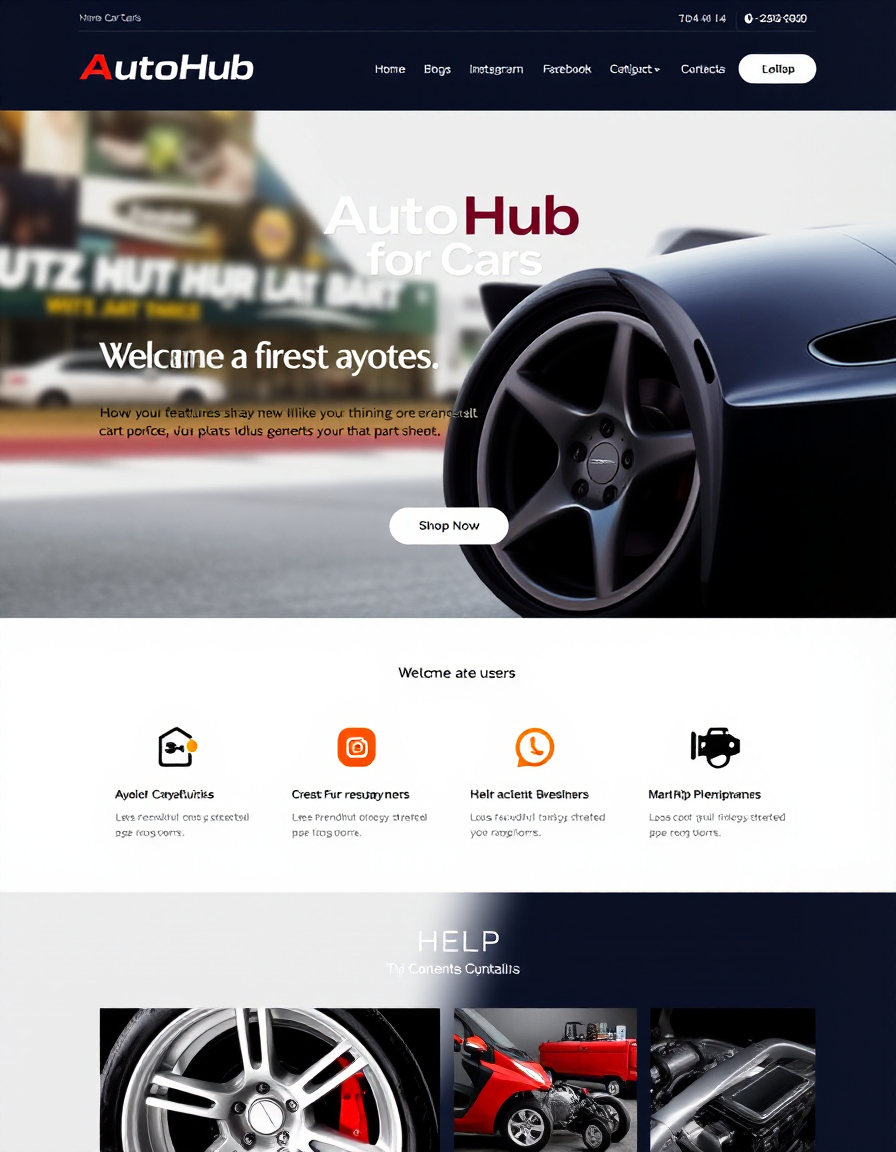
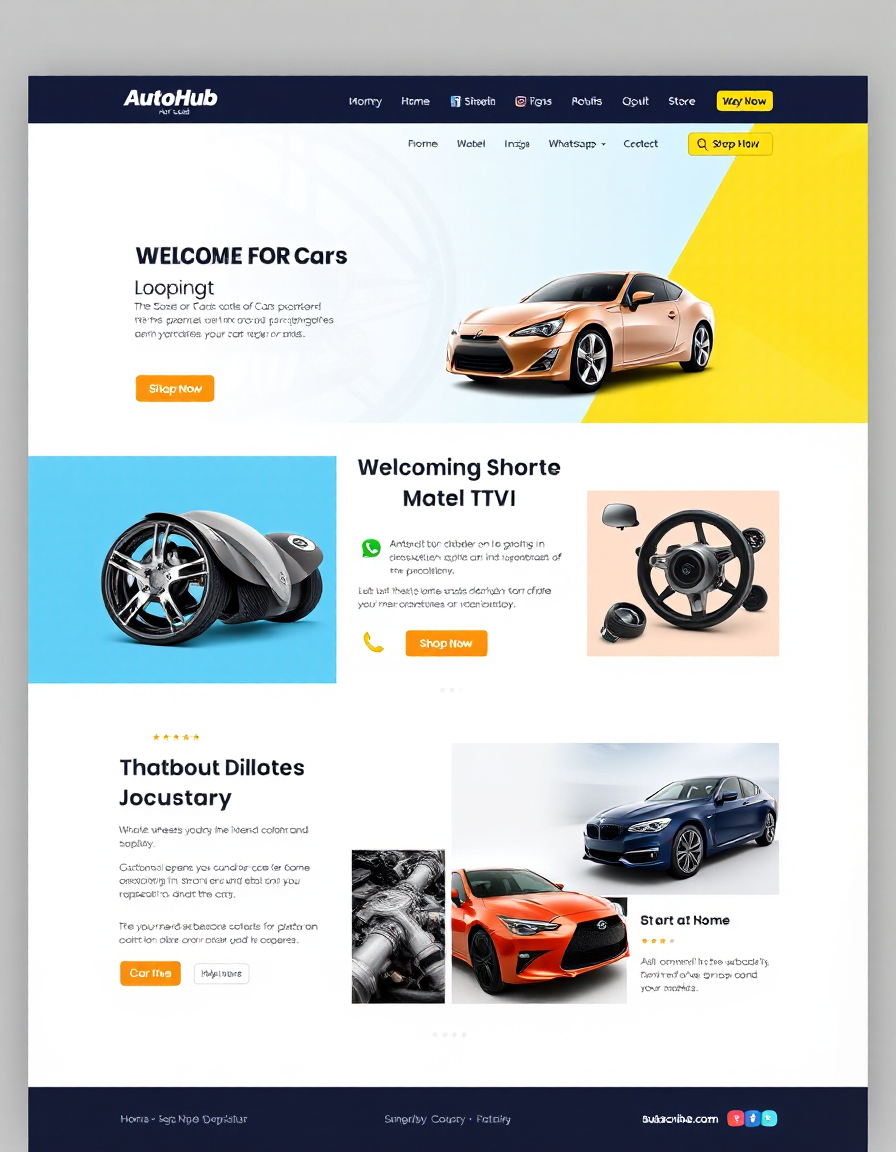
* Actors: Dealer, Frontend, Backend, Database.
* Steps:
* Dealer logs in and navigates to the "List a Car" page.
* Dealer enters car details (e.g., brand, model, price) on the frontend.
* Frontend sends the car details to the backend.
* Backend validates the data and stores it in the database.
* Backend sends a confirmation to the frontend.
* Frontend displays a success message to the dealer.
* **Sequence Diagram for Dealer Lists a Car:**

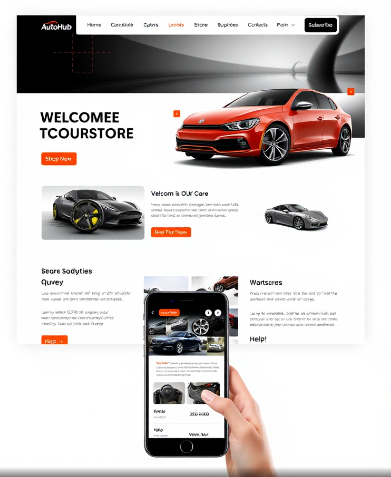
****

## **Class Diagram:**



## **Prototype :**





# **AutoHub" Project Implementation Plan**

## **Selection of Tools and Programming language :-**

The programming languages and tools used in the project were selected considering the requirements of the project, which also included some technical and practical characteristics, as depicted in the following table:

|  |  |  |
| --- | --- | --- |
| Component | Technology/Tool | Justification |
| Frontend | React.js | To create interactive, modern interfaces that are reusable with ease of management. |
| UI Design | Tailwind CSS / Bootstrap | For designing consistent and responsive interfaces to enhance the user experience. |
| Page Structure | HTML and CSS | To construct the overall skeleton of the interface, format the content in an attractive and easy-to-use manner. |
| Backend | Node.js with Express.js | For rapid request processing, this lightweight framework provides a flexible way to organize services. |
| Database | PostgreSQL | handling of huge volumes of data and relationships between tables; high performance and stability expected. |
| AI | TensorFlow / Scikit-learn | Develop the intelligent search based on user preferences and recommendations of products. |
| Version Control | Git with GitHub | This will track the changes within the code and maintain the process of team collaboration effectively and organized. |

**Justification**:

* Community Support: Communities contribute to these products with support and documentation.
* Sustainability: These tools can be agile, scalable, as the platform is growing.
* Cost: The tools fit within the estimated budget for this project.

## **Programming Standards**

The project is designed on a set of strict standards to ensure the code is of high quality and maintainable. These are as follows:

### **Naming Conventions**

* Variables: The variable names are clear and descriptive, such as userProfile and carDetails.
* Functions: The functions are in CamelCase format, e.g., getUserDetails and processPayment.
* Constants: Constants are written in uppercase with words separated by underscores, e.g., MAX\_RETRIES and API\_URL.

### **Comments**

* General Comments: The purpose of each module is explained.
* Detailed Comments: Complex operations within the code are explained. Example:  
  javascript  
    
  // Database query to retrieve information about the user  
  async function getUserDetails(userId) {  
  // Connect to database to fetch user details  
  }

### **File Structure**

The files are structured in this manner in the project:

* models: Will contain data models, e.g., table definitions.
* controllers: Will handle the requests and perform actions.
* routes: API routes definition.
* public: This will hold your public files, e.g., images, static assets.
* views: Contains frontend files.

### **Indentation**

ESLint and Prettier are among those that ensure code has the same feel and look for all developers on a team.

## **Version Control**

Version Control Tools

* Git: This will be the tool to implement for version control; thus, keep track of every single modification made on the code.
* GitHub: A centralized repository using GitHub will be maintained. The following are the features needed for GitHub:
* Branches:
* The main branch : main should be where the code is stable
* Feature Branches: A new branch is created for every feature. Suppose: feature/smart-search.
* Pull requests: To review the code before merge.
* Documentation: CHANGELOG.md will reflect all the updates.  
  Practical Procedures

|  |  |
| --- | --- |
| Step | Description |
| Create a New Branch | A new branch is created for each feature addition or modification. |
| Apply Changes | Changes are made in the local, and then those changes are pushed to the repository by using git push. |
| Code Review | Pull request code is reviewed by other team members. |
| Merge the Branch | When the pull request is approved, that branch is merged into the main branch. |

## **Software Modules**

### **Detailed Software Modules:**

The project has been divided into smaller modules so that the development and maintenance of the project can be handled easily, as stated below:

|  |  |
| --- | --- |
| Description | Module |
| Concerned with User Registration, Log-in, and Account Handling. | User Management Module |
| Add/Update/Delete the car listing; view the detail along with Images. | Car Management Module |
| AI-based search for a car or a spare part or recommendation. | Smart Search Module |
| Keeps records regarding spare parts along with compatibility among different brands. | Spare Parts Management Module |
| Deals with all the Online Payment and Subscriptions. | Payment Module |
| Instant notifications concerning offers or updates to the user. | Notification Module |
| A module containing a personal interface for merchants in managing listings and viewing statistics. | Dashboard Module |

### **Importance of Modularity**

* Ease of Maintenance: The modules are independent in nature, updating without affecting the whole system.
* Reusability: Modules can be used in other future projects.
* Scalability: Addition of new modules can be added easily.

## **Conclusion**

The above-execution plan assures observation of the quality and organizational standards and forms a very realistic and holistic framework for developing the "AutoHub" project. The core idea behind using the best set of tools and technologies forms part of this working module for effective and easy maintainability.

# **Deployment Strategy**

## **Environment Setup**

The environment setup ensures that the production environment is ready to host the platform, including server setup, database configurations, and integrations with third-party services.

### **Server Setup:**

* **Type:** Cloud-based server, such as AWS EC2, Google Cloud, or Azure VM.
* **Specifications:**
* CPU: 4 cores or higher.
* RAM: 8GB or higher.
* Storage: 100GB SSD, scalable based on the growth of data.
* Operating System: Ubuntu 20.04 LTS (Linux).
* **Steps:**
* Provision a cloud server instance.
* Configure firewall rules to allow HTTP (port 80), HTTPS (port 443), and SSH (port 22) traffic.
* Install necessary system dependencies (e.g., Git, Node.js, Nginx).

### **Database Configuration:**

* **Type:** Relational database (e.g., MySQL or PostgreSQL).
* **Configuration**:
* Database Name: car\_spare\_parts\_db.
* User: admin\_user with full privileges.
* Password: Secure password (e.g., 12+ characters with special symbols).
* **Steps**:
* Install MySQL or PostgreSQL on the server.
* Create the database and user with the necessary privileges.
* Import the initial database schema and data.

### **Web Server Setup:**

* **Type:** Nginx or Apache.
* **Configuration:**
* Ports: 80 - HTTP, 443 - HTTPS.
* SSL Certificate: An SSL certificate, like Let's Encrypt, to securely communicate.
* **Steps**:
* Install Nginx or Apache on the server.
* Configure the web server to serve the frontend and backend.
* Configure SSL for HTTPS.

### **Backend Setup:**

* **Runtime Environment:** Node.js-16.x or higher.
* **Dependencies:** Run npm install to install all the required packages.
* **Steps**:
* Clone the repository of the backend from GitHub.
* Install Node.js and dependencies.
* Set environment variables, such as database credentials and API keys.
* Start the backend server by using any process manager, for example, PM2.

### **Frontend Setup:**

* **Framework:** React.js (version 18.x or higher).
* **Dependencies**: Install the required packages by running npm install.
* **Steps**:
* Clone the frontend repository from GitHub.
* Install Node.js and dependencies.
* Do a build of the frontend using npm run build.
* Serve the built files using Nginx or Apache.

### **Third-Party Integrations:**

* Payment Gateway: Configure API keys to process the payments on the website via PayPal or Stripe.
* AI/ML Services: Set up AI libraries like TensorFlow and Scikit-learn to use smart search functionality.

## **Installation Guide**

This document provides the installation and execution steps needed by the software-developers and users.

### **For Developers**

* Clone the repository:
* Backend: git clone <https://github.com/your-repo/backend.git>
* Frontend: git clone <https://github.com/your-repo/frontend.git>
* Install Dependencies:
* Backend: Navigate into the backend folder and type npm install.
* Frontend: Navigate into the frontend folder and type npm install.
* Environment Variables Setup
* Backend: Create a.env file and include the following variables:  
  Copy
* DB\_HOST=localhost  
  DB\_USER=admin\_user  
  DB\_PASSWORD=your\_password  
  DB\_NAME=car\_spare\_parts\_db  
  PAYMENT\_API\_KEY=your\_payment\_api\_key
* Frontend: In the root, create a.env file with the following variable:  
  Copy  
  REACT\_APP\_API\_URL=http://localhost:5000
* Run Application:
* Backend: To start the application, run either npm start or node server.js.
* Frontend: Start the application by running npm start.
* Access Application:
* Open the browser and go to http://localhost:3000 for the frontend.
* The backend will be reachable at <http://localhost:5000>.

### **For End Users:**

* Web-Based Access:
* The users can access the platform through the web.
* URL: https://www.autohub.com

## **Deployment Strategy**

The deployment strategy expounds on the delivery of the software to users and how its updates will be managed.

### **Web-Based Deployment:**

#### **Continuous Integration/Continuous Deployment (CI/CD):**

* Utilize the CI/CD tools such as GitHub Actions, Jenkins, or CircleCI for automating deployment.
* Create pipelines that automatically deploy the changes in code to the production server.

#### **Version Control:**

* Use Git for version control, with different branches for development, staging, and production.
* Only merge code into the production branch after rigorous testing.

#### **Rolling Updates:**

* Progressively roll out deployments to limit the impact on uptime and reduce risk.
* Leverage Load Balancers: direct traffic onto updated servers.

#### **Monitoring and Logging:**

* Implement monitoring tooling-such as New Relic and Datadog-to observe performance and errors in a system.
* Logging tooling-ELK Stack, Splunk, and similar-provide a means of collecting and processing logs.

### **Packaged Software (Optional)**

#### **Docker Containers:**

* Wrap the application into Docker containers for easy deployment and scaling.
* Manage multi-container deployments with Docker Compose-frontend, backend, and database, for instance.

#### **Kubernetes Optional**

Use Kubernetes for container orchestration and scaling in the production environment.

# **Project Management and Planning**

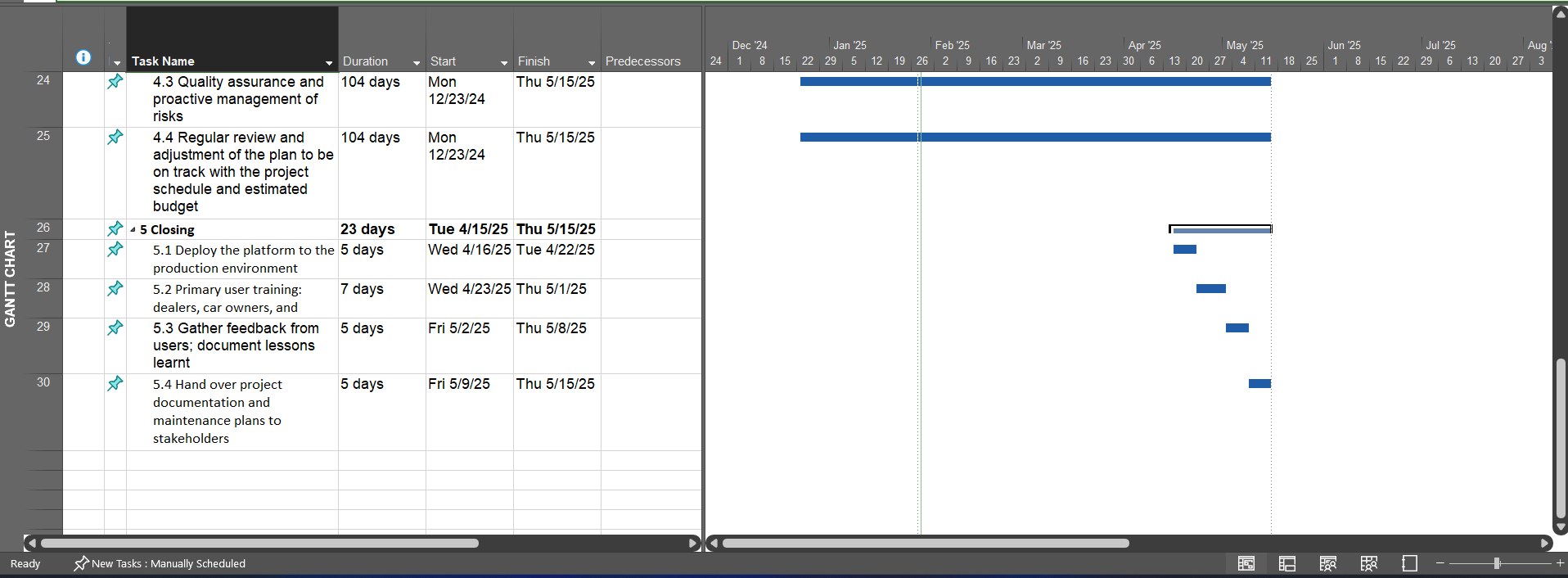
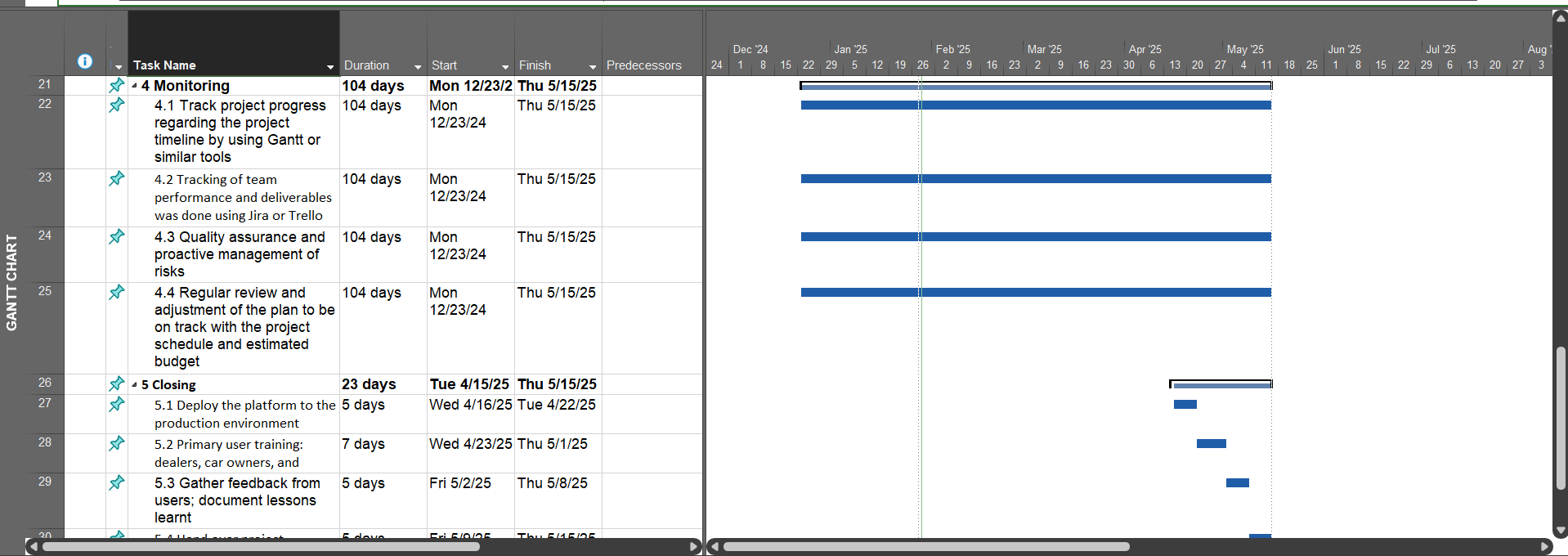
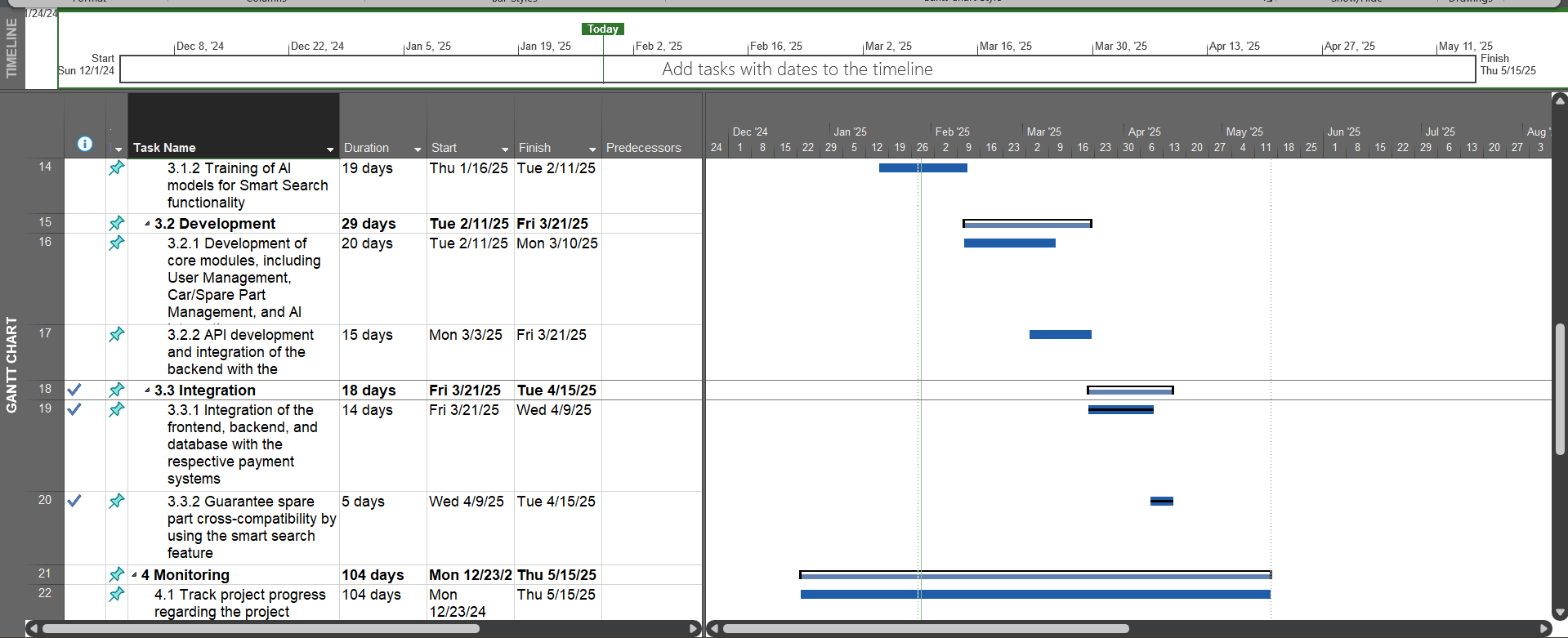
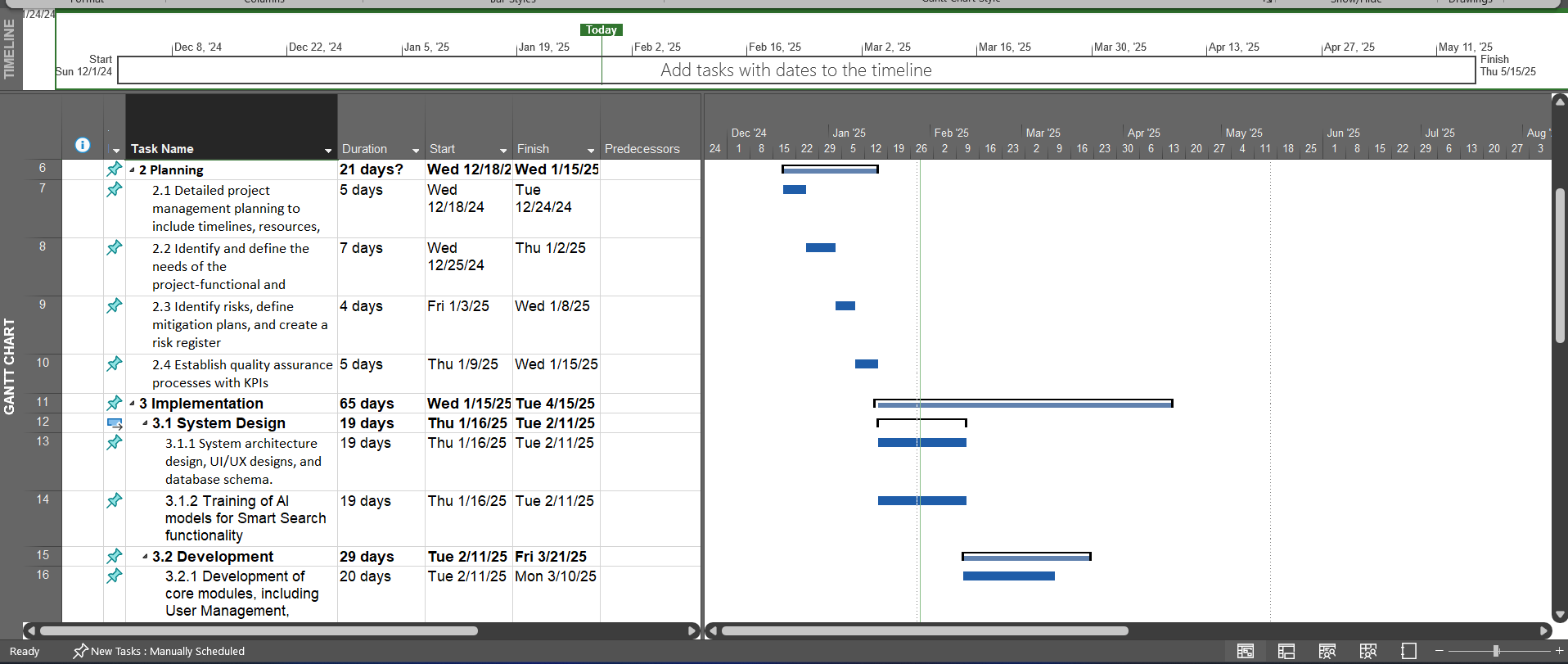
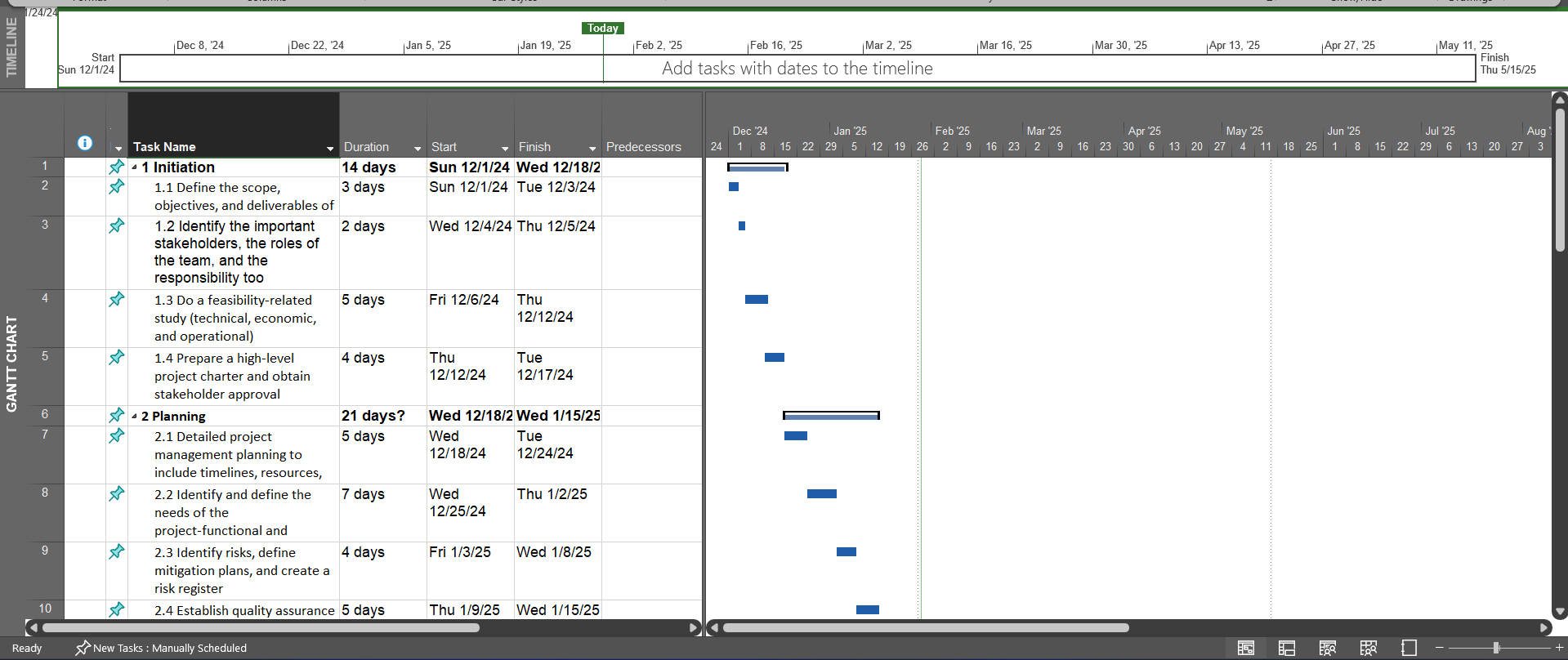
## **Work Breakdown Structure (WBS)**

The entire project management is divided into five major phases, each comprising certain tasks that would help in completing the process in a sequential manner:

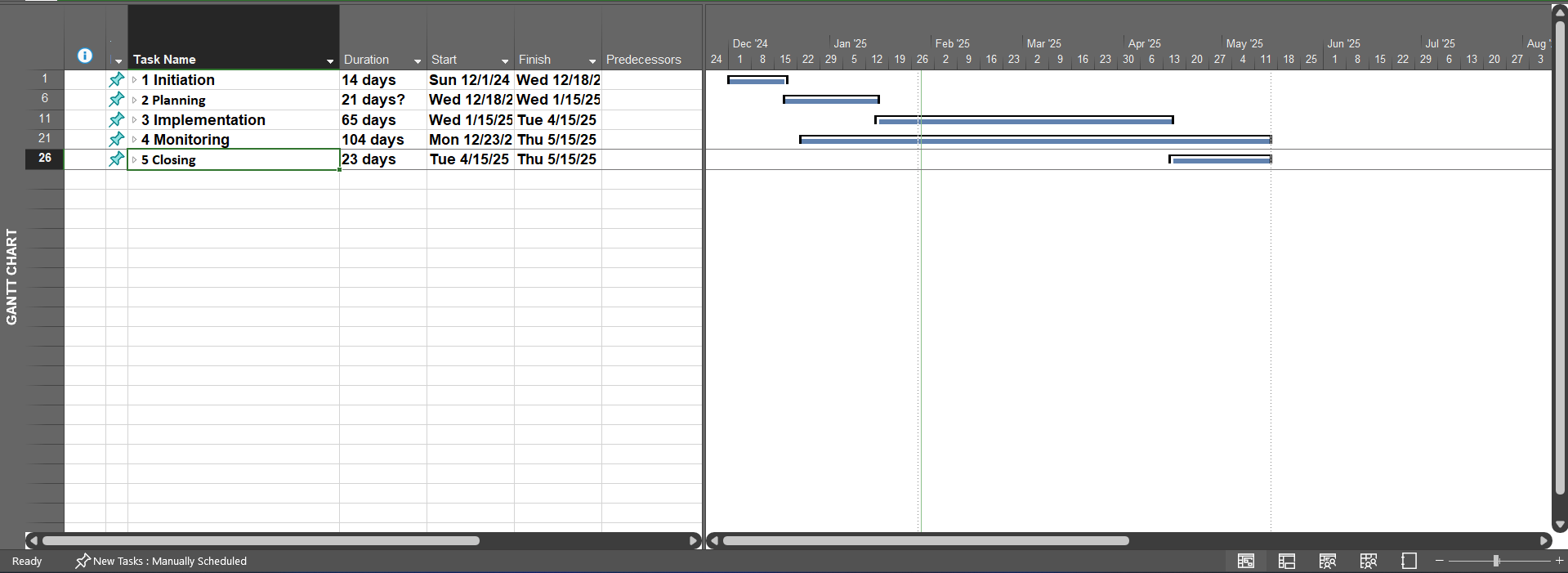
Phases:

1. Initiation
   1. Define the scope, objectives, and deliverables of the project.
   2. Identify the important stakeholders, the roles of the team, and the responsibility too.
   3. Do a feasibility-related study (technical, economic, and operational).
   4. Prepare a high-level project charter and obtain stakeholder approval.
2. Planning
   1. Detailed project management planning to include timelines, resources, and budget.
   2. Identify and define the needs of the project-functional and non-functional.
   3. Identify risks, define mitigation plans, and create a risk register.
   4. Establish quality assurance processes with KPIs.
3. Implementation
   1. System Design:
      1. System architecture design, UI/UX designs, and database schema.
      2. Training of AI models for Smart Search functionality.
   2. Development:
      1. Development of core modules, including User Management, Car/Spare Part Management, and AI Integration.
      2. API development and integration of the backend with the database.
   3. Integration:
      1. Integration of the frontend, backend, and database with the respective payment systems.
      2. Guarantee spare part cross-compatibility by using the smart search feature.
4. Monitoring
   1. Track project progress regarding the project timeline by using Gantt or similar tools.
   2. Tracking of team performance and deliverables was done using Jira or Trello.
   3. Quality assurance and proactive management of risks.
   4. Regular review and adjustment of the plan to be on track with the project schedule and estimated budget.
5. Closing
   1. Deploy the platform to the production environment.
   2. Primary user training: dealers, car owners, and workshops.
   3. Gather feedback from users; document lessons learnt.
   4. Hand over project documentation and maintenance plans to stakeholders.

## **Timeline:**



And in this screen shot the Gant chart for main process for project



## Roles and Responsibilities

The role of each team member will be clearly defined so as to ensure orderly accomplishment but still conform to the objective of the whole project:

|  |  |  |
| --- | --- | --- |
| Role | Team Member | Responsibilities |
| Project Manager | Mohammad Atteyah | Oversee all phases; ensure timelines are met, manage risks, and coordinate teams. |
| Business Lead (CEO) | Baseem Al-Jaghoob | Align project execution with the business goals, communicate to stakeholders, and ensure proper resource allocation. |
| Software Engineering | Ibrahem Atatri | Refine and analyze system requirements to ensure feasibility and alignment with project goals. |
| Frontend Developer | Ya'qoob | Create responsive, accessible user interfaces and integrate backend services. |
| Backend Developer | Ahmad | Develop server-side logic,APIs, and database operations. |
| AI Specialist | Jameel | Develop AI-powered smart search algorithms and machine learning model training. |

Collaboration Guidelines:

* Use GitHub for versioning code; review all features via pull requests.
* Weekly team meetings to review progress, address any issues, and update tasks accordingly.
* Documentation of all decisions, updates, and changes.

**Conclusion**

The AutoHub project is one of these transformational projects that aim to solve some of the big inefficiencies and challenges of the Palestinian automotive marketplace. By seeking to develop an inclusive, easy-to-use, advanced technological web-based platform, AutoHub is bound to alter the way in which car owners, dealers, and service providers communicate, carry out business with, and take care of their vehicles. Smoothen the core focus on the transaction for an improved user experience, add pioneering features like AI-powered Smart Search, Cross-Compatibility databases, safety in payment via platform cuts, and seal the deal for game-changing status in the local market.  
  
Key Achievements and Highlights:

1. Problem-solving: Till today, according to AutoHub, there has never been one single efficient digital marketplace for buying, renting, and sourcing of spare parts in Palestine. It does away with the exploitative practice of saving the time of users, hence their confidence to perform transactions with ease.
2. It is an innovation that does not stop at advanced technologies powering AI-driven algorithms in the search capabilities, cross-compatibility databases, or a real dealer dashboard, for that matter, at AutoHub-what really makes it different from every other hub ever built. Such features give better experiences to users while stronger tools are dispensed to the dealers and service providers for efficient management of businesses.
3. Robust System Architecture: AutoHub is based on a three-tier architecture: front-end, back-end, and database. To this end, modern technologies such as React.js, Node.js, and PostgreSQL will be used for the proposed system to guarantee top performance, scalability, and security.
4. Feasibility and Economic Viability: The feasibility study through deep financial analysis shows that the AutoHub is indeed technically feasible and economically viable. Equipped with different revenue channels like merchant subscriptions, transactional fees, and advertising, the project holds great promise financially.
5. User-Centric Design: An easy-to-use interface which AutoHub is going to provide, supplemented with smart search functionality and secure forms of payment, will make the website manageable for all users, irrespective of their backgrounds. Besides, user reviews and ratings add to the trust, ensuring much more transparency in marketplace dealings.
6. This gives the system quite a large leeway toward scaling and any foreseeable future expansion or growth of development in the future, making this platform an optimal solution that matches the market expansion and addition requirements for more integrations with other services and possible other markets as well.

**Impact in the Palestinian Market:**

AutoHub will strongly impact the field of Palestinian Automobiles through the following aspects of:

* Efficiency Enhancement: Time and effort spent in purchasing or selling cars, spare parts sourcing will be minimized.
* Empower the Users: To offer a credible and transparent platform to car owners, dealers, and service providers for conducting their business.
* Increase Economic Activity: Seamless and easier transactions reduce frictions which, in turn, will lead to growth in the local automotive industry through AutoHub.
* Encourage Digital Adoption: This encourages the digital way in traditionally offline markets, creating room for more use of technology.

Although this is a highly potential project, there are challenges like new entrants, delays in import services, and continuous updates required to keep the platform relevant. In mitigating such risks, some strategies which AutoHub would employ include:

* Continuous Cycle Improvement: The site should be renovated day by day; adding and developing new features by implementing the received feedback from the users.
* Strong Supplier Networks: The relationships with suppliers abroad will contribute a lot in spare part delivery at the right time.
* Digital Marketing and Customer Engagement: Digital marketing will target the customers of service by attracting and retaining customers with offer and campaign advertisements.

**Final Words:**  
The AutoHub project has been one of the success stories in innovative solution development, applying and integrating more recent technologies. AutoHub fills the needed gap within the Palestinian automotive marketplace and hence offers fantastic service to its users, as well as a new bar toward which other digital platforms aim in the region. A sound foundation, well-focused vision, and an adequate team place AutoHub at the best possible place to meet their goals and be able to engrave a real mark in the automotive sector within Palestine.  
  
It means the commitment of the team to creating a qualitative platform that can meet the needs of its end-users, while observing the highest standards related to security, usability, and performance. Successfully implemented, AutoHub would be an important milestone in the journey of the Palestinian market to complete its path toward digital transformation and open ways for more innovation and growth afterward.