PART PRO MANAGEMENT SYSTEM

Group Name: Part Pro Management System

Group Members:

1. Sai Rahul Padma - sairahulpadma@my.unt.edu -11714588
2. Nikhita Muvva - NikhitaMuvva@my.unt.edu- 11601586
3. Sneha Reddy Gangannagari - snehareddygangannagari@my.unt.edu -11642761
4. HimaBindu Chunduri - himabinduchunduri@my.unt.edu -11724307
5. Pavani Venigalla - PavaniVenigalla@my.unt.edu -11697604
6. Sai Samyuktha Paspuleti - SaiSamyukthaPaspuleti@my.unt.edu -11654305
7. Jhasha Sri Ede - JhashaSriEde@my.unt.edu -11685874
8. Rishika Yalamanchili - RishikaYalamanchili@my.unt.edu -11685270

PROJECT DESCRIPTION:

A Part Pro Management System website is used to buy vehicle related parts and products. This application will be used by customers and employees. Customers will add their vehicle related details and the relative products will be dynamically displayed in the customized dashboard page. User will checkout the products and complete the payment through payment gateways. Employees will use the application to add products in their inventory by selecting available suppliers and available products from the suppliers. Employees can view and manage the online sales.

Major Functionalities:

1. Customers can search for products and dynamically products and related categories will be shown in the dashboard. Different types of categories will be available and all the available products in the store will be populated once user lands on dashboard page.
2. Customer can view all the product related details by clicking on each product. He/She can see the availability of the product, product description, customer rating etc.
3. Customer's current location will be fetched and all the nearest store locations will be shown to the customer. The nearest store location will be displayed on the customer's dashboard page.
4. Customers can check out the products and can view or manage all the product details in cart page. User can complete the payment once user verifies all the product detail in cart page.
5. User can complete the payment through payment gateway and user can view the confirmation details regarding the payment they have made. If the payment was successful user can view the acknowledgment page, and If payment was declined or pending, warnings will be shown in a UI banner.

6. Multiple locations will be available for a store and products will be stalked and multiple types of products will be available in an inventory. Product Quantity and product status like 'Out of Status', 'Available' etc., can be viewed for each product in the specific location.

7. Different types of roles assignments like Store Owner, Assistant Employee, Associate Employee etc., will be managing the entire store. Store owner will able to manage entire store and will have access to online and offline sales. Associate or Assistant Employees will assist the customers for the orders and payments done by them.

8. Customers and Employees can sign up for the website and login into the application. Authentication will be done in two steps. Users will login into the application by entering username and password. Once user was able to authenticate successfully, Multi Factor authentication will be done by entering the OTP which was sent to the user's email ID. Payment capabilities were also integrated using third party payment gateways.

9. Users will login securely into the application and view all the product details and checkout the required products and complete the payments. The entire process will be simple, smooth and dashboard is customizable as users search for the products and select the required.

10. The application is responsive and it is user friendly for both customers and employees. The application is cloud native and cloud agnostic. The application can be used in real time as it can be scaled up or scaled down based on the web traffic as the web application is deployed in Azure VM cloud.

11. Employees can select all the available suppliers who supplies products to the required locations. Multiple suppliers would have capability of supplying multiple products to each of the locations. Once employee acknowledges the products which needs to be added in the Inventory, inventory will be updated with product quantity and status.

12. Store owner can view and manage all the online sales done by the customers. He/She can view the total price, which product has been sold, which customer has purchased and which employee got engaged to assist the customers.

Software Requirements:

Software Used: Eclipse, JDK, Postman, Node, Maven, Visual Studio Code, Putty, Beyond Compare, GitBash, MySQL.

Supported Browsers: Google Chrome, Safari and Microsoft edge.

Frontend Stack: HTML, CSS, JavaScript , Bootstrap, React js.

Backend Stack: Core Java, Spring Boot, JPA, MySQL and Docker.

Software Versions:

Java JDK Version: 21

Maven Version: 3.8

Node Version: 14.2

Docker Version: 24.0.5, npm Version: 6.14.18

PROJECT TIMELINE:

Tasks	Start Date	End Date
Project Planning and Finalization	16/01/2024	23/01/2024
Project Design	23/01/2024	30/01/2024
Documentation for Deliverable 1	30/01/2024	5/2/24
Presentation for Deliverable 1	30/01/2024	5/2/24
UseCase Diagrams and Database Planning	6/2/24	12/2/24
Initial Cloud Setup	12/2/24	19/02/2024
Documentation for Deliverable 2	19/2/2024	27/02/2024
Backend Development	28/02/2024	18/03/2024
API Testing	28/02/2024	18/03/2024
Documentation for Deliverable 3	28/02/2024	18/03/2024
Frontend Development	19/03/2024	8/4/24
UI Testing	19/03/2024	8/4/24
Documentation for Deliverable 4	19/03/2024	8/4/24
Cloud Deployment	9/8/24	15/04/2024
Regression Testing	9/8/24	15/04/2024
Integration and Performance Testing	16/8/2024	22/04/2024
User Acceptance Testing	16/8/2024	22/04/2024
Documentation for Deliverable 5	23/04/2024	29/04/2024
Presentation for Final Deliverable	23/04/2024	29/04/2024

The Gantt chart columns span dates: 16-Jan, 23-Jan, 30-Jan, 5-Feb, 6-Feb, 12-Feb, 19-Feb, 27-Feb, 28-Feb, 4-Mar, 11-Mar, 18-Mar, 19-Mar, 25-Mar, 2-Apr, 8-Apr, 9-Apr, 15-Apr, 16-Apr, 22-Apr, 23-Apr, 29-Apr.

RISK MANGAMENT:

Risk 1:

Uncaught and Caught Exceptions Risk: When we are working on the back end, there will be multiple exceptions thrown by java run time. It will throw 404 and 400 bad requests.

Reevaluation:

1. It is crucial to handle all kind of exceptions and include all the service layer calls in try catch block mechanisms.
2. Need to write unit test cases with negative testing to catch and identify all the exceptions.

Contingency Plans:

1. Need to create custom exception handling layer to avoid any java run time errors.
2. Need to test all the back-end APIs before the code gets deployed.

Risk 2:

CORS Connectivity Issues and Cloud Configurations Risk: As back-end application is stand alone and it runs on embedded tomcat server, front end needs to be pointed to back-end REST API's. So, as they both run on different ports, it would have the risk of cross origins.

Reevaluation:
1. Review the CORS configuration once and twice before the application gets deployed.
2. Determine and Review which other ports needs to be exposed i.e., which depends on the application . Add all the ports in security target groups in Azure VM configuration to avoid network errors and internal server errors.

Contingency Plans:
1. CORS configuration related code needs to be added in back-end packages.
2. All ports need to be public in cloud Azure VM to avoid connectivity and CORS errors.

Risk 3:

Dependency and Third-Party Integration Failures: As the application is deployed in cloud, there will be network issues and the application will throw us internal server errors. If the application has connectivity issues with third party systems, it will throw gateway time out errors.

Reevaluation:
1. Monitor the health of the third-party integration systems like payment gateways and Identity Access Management etc.
2. Regularly review the reliability of the system and third-party libraries.

Contingency Plans:
1. Develop fall back and retry mechanisms for all the services in the back end.
2. Monitor the network and connectivity logs in azure cloud server.

TEAM ROLES:

1. Project Lead – Nikhita Muvva

2. Design Lead - Sai Samyuktha Paspuleti

3. Requirements Lead - Jhasha Sri Ede

4. Lead for Front End – Nikhita Muvva

5. Lead for Back End - Sai Rahul Padma

6. Configurations Lead – Sneha Reddy Gangannagari

7. Testing Lead – HimaBindu Chunduri

8. Documentation Lead – Sai Rahul Padma

9. Presentation Lead – Pavani Venigalla

10. Devops Lead – Rishika Yalamanchili

MEMBER CONTRIBUTION TABLE:

Member name	Contribution description	Overall Contribution (%)	Note (if applicable)
Sai Rahul Padma	I have worked on architectural design and explained idea to the team.	13.5	
Nikhita Muvva	I have done the entire project planning and divided the tasks.	13	
Sai Samyuktha Paspuleti	I have worked on preparing timeline charts along with Hima Bindu and prepared initial GitHub repository structure.	12.5	
Rishika Yalamanchili	I have worked on documentation part in this deliverable along with Sneha.	12	
Pavani Venigalla	I have worked on presentation part and involved in risk management identification.	12.5	
Himabindu Chunduri	I have worked on preparing timeline chart for this project and prepared initial GitHub repository structure.	12	
Sneha Reddy Gangannagari	I have worked on documentation in this deliverable along with Rishika.	12.5	
Jhasha Sri Ede	I have worked on presentation part and involved in risk management identification.	12	