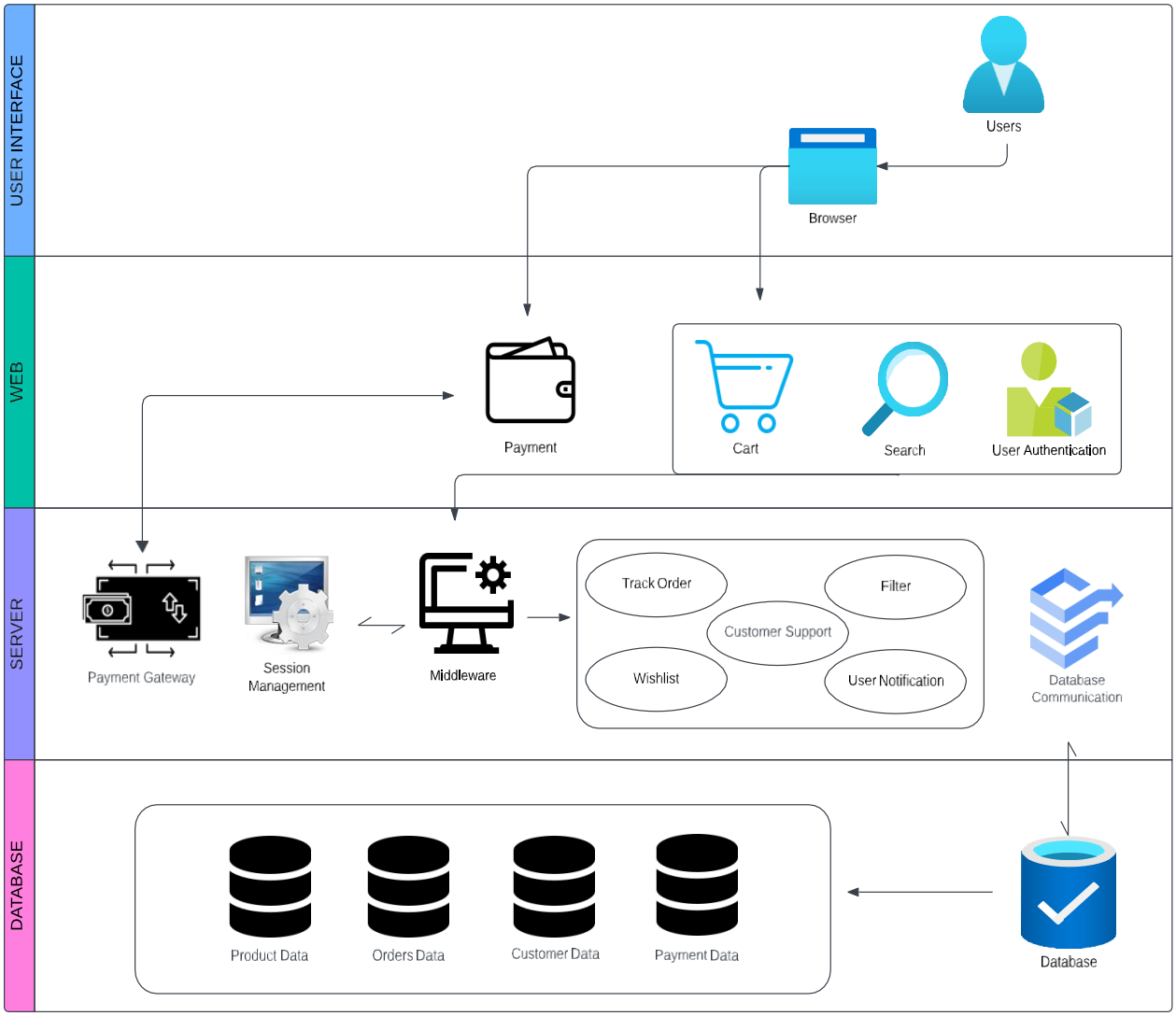
Project Name : Toy Station
Group-3

System Structure:



TOP-LEVEL DESCRIPTION FOR THE SYSTEM:

1**. User Interface Layer:**

This is the layer where the user communicates with the application; it is the first layer of the application.

- **Users:** Anyone who has used the website or application at hand as a client or a

consumer. They use their browser to navigate through products, search, purchase products and even monitor their orders.

- **Browser:** The browser works in that it is the client-side component that allows users to access the system. It becomes the mean of announcing requests to the web server like searching for some products or performing payments and displaying results such as information about the products and other services.

2. Web Layer:

This layer comprises the elements which are accessed through the web browser by the users of the system. It is positioned between the GUI and the server layer.

- **Cart:** It carries products that the user requires for purchase. The list of products to be purchased is stored in the cart but customers are able hold, change or remove items in it until the order is complete.
- **Search:** A interface where users can enter a query and get results from a set of products or services on the platform. These send a request to the server to go and search for matching products in the database.
- **User Authentication:** Regarding Information Technology, this is the system that compels users to enter their identity facts such as usernames, passwords, or other identification and physical verification like two-factor authentication systems. It makes sure only the permitted people use particular services such as account information and payment.
- **Payment:** It handles the initiation of payment in this module. Whenever the user tends to confirm his/her cart and make a purchase, this component collaborates with the server for making the payment through Payment Gateway and also interacts with Payment Data in Database.

3. Server Layer :

This layer contains the actual computational functionality of the system and performs a middleman responsibility of relaying messages between the web layer and the database layer.

- **Payment Gateway:** The payment gateway then initiates an interaction with external payment systems (credit card merchant, PayPal, etc.), to process the payment. It makes a confirmation of the security of the given transaction and the successful execution of the payment made by the user.
- **Session Management:** This service preserves a client's session – information like login status, items in a cart are maintained throughout the duration of the user's interaction with the site. It follows up the users and records every activity they undertake and every preference when they are undergoing through the system.
- **Middleware:** Middleware is a core concept that deals with request handling, manipulating and directing such requests from the web application to the server. It integrates payment, cart, search and even authentication services all into one to make them work cohesively.
- The middleware connects to several important services:
- **Track Order**: The general purpose of this feature is to allow consumers to access information concerning the status and position of an order they made.
- **Customer Support:** Supports users to solve problems or reply to questions they may have concerning the items they ordered or services they bought.
- **Wishlist:** Enables user's undertake purchase or view products at a later time.
- **User Notification**: Delivers messages (e.g., orders confirmation, delivery information) to the customers.
- **Filter**: Aids a consumer into making wiser purchasing decisions through offering a filter mechanism such as by the price, brand or rating.
- **Database Communication**:This service helps to make communication between the server and the database fluent to get and store data like product information, customers' data or orders.

4. Database Layer :

This functional area is where all the valuable business information is processed and archived. It contains all kinds of information required in the system, storage of which is imperative for any

computer operation.

- **Product Data**: It holds all details and specifications connected with the products including the descriptions, price, availability, images and category. This data is pulled each time a user shops by searching or by browsing through the available varieties of products.
- **Orders Data:** Tables the facts about every order conducted by users like their included items, payment status, shipment, and delivery information.
- **Customer Data:** Contains customer's account data, orders, addresses, and credit cards.
- **Payment Data:** This holds information concerning payment options for the users including credit card information, favorite payment method, and the numeric history of payments made online. It operates most closely with the payment gateway to authenticate the purchases made.
- **Database:** Through the database however all the above data is stored and transferred to the server. They process read, write, update, and delete request where the system obtains that current information as required in the user interface or the server.
- This layered architecture makes the system organized and more easily scalable and maintained and free from the problem of complication through the separation of concerns by layers.

Requirements Specifications:

Functional Requirement:

Functional Requirement	Description	Example
Sign up	Creating Account with username and password fulfilling all required information.	Full Name, Email Address, Phone Number, Password, Checkbox for Terms and conditions.
Login	Login with your created account credentials	Username: xyz@gmail.com Password: abcd12@
Search Product	It exactly searches according to your requirements.	Soft toys, baby toys, remote controlled toys, gift toys, puzzles
Filters	Viewers can filter products according to their needs.	age, brand logos, color, discount, price high to low or low to high, offers, new arrivals
Add to Cart	Customers can add all the selected toys to their shopping cart.	Adding your selected toy to your cart to purchase.
Wishlist	Users can save their favorite toys on the Wishlist.	You can add toys to favorite list to buy.
Checkout	Displays your cost for the order, asks your shipping address, payment method and delivery instructions, and the estimated arrival is shown.	Checkout helps users to show the entire details of product confirmation.

Payment Portal	It asks for your credit or debit card details like card number, CVV and expiry date.	Credit card No: 16479695 Cvv:346, Expire day: 08/28
Track your order	Customers can track their order in orders placed in their profile.	Tracking Id: 7978270989078 Location: Denton, TX
User Notifications	Customers get order confirmation emails, order tracking and delivery text messages.	Processing, Shipped, out for delivery.
Customer Support	Customer support system has an ability to chat with the user for many problems and questions (with the help of the artificial intelligence that answers in the night and day). This assists in addressing many of the arising questions without involving the human aspect hence providing support at any time.	A user may say to a chatbot, "What is the return policy?" and get an instant, pre-populated, and system-generated message containing all the pertinent information they need.

Non-Functional Requirement:

Non-Functional Requirement	Description	Example
Productivity	It should give optimal outputs even during peak times	Website must load within 3-5 secs even during high time.
Connectivity	It should connect huge number of people without any network issues.	It must handle 5000-7000 customers at the same time without any performance breakdown.
Security	All the user details should be encoded.	especially personal details and user payment details
Compatibility	Website should be accessible to any user around the world at any time in any device (iOS or android).	May be accessing with iPhone or Windows Laptop.
User Friendly Interface	It should be easily understood and be used by the user without any technical knowledge.	User should easily differentiate without any ambiguity.
Data Storage	Every detail of data that is being produced should be securely stored.	Data backup, User data, and Transaction History should be stored.
Multi-Language Maintenance	User should be able to change their language according to their region for better experience.	English, Spanish, Germany, Japanese, French, etc.…
Currency Check	Customer should be able to process payment and view the cost of the toys based on the country they live.	Dollars, Euros, Rupees, Pounds, Dirham

Children Safety	Special precautions should be taken against children viewing insensitive content by not redirecting and displaying unrelated content	Adult content, Violence
Laws	The application must check all the e-commerce and customer tax regulations.	Sales Tax, VAT, GST, DST

Interface Requirements:

Interface Requirements		**Description**	**Example**
User Requirements	Customers	Ease of navigation, security of payment options, ease of checkout, and mobile compatibility.	Easy navigation, checkout via Shopify/WooCommerce.
	Admin	A dashboard that shall be used to administer inventories, track orders, and manage users.	This is where you'll do things like manage your products and orders in the Shopify Admin.
	Developers	They can extend and scale by offering APIs to integrate new functionalities.	Developers Extend store features using the WooCommerce API.
Hardware Requirements	Servers	Host reliably with load manageable cloud hosting services like AWS and Azure.	Also host on AWS or DigitalOcean for scaling.
	User Devices	Customer accessing the site from laptops, desktops, tablets, and smartphone	Desktops, tablets, and smartphones.

	Systems for Backups	For data redundancy, use an external server or cloud backup.	Use UpdraftPlus to backup your website automatically to the cloud.
Software Requirements	CMS	Easy-to-use Content Management System for managing product listings, pages, and updating content.	WordPress- for blog and product management, or Wix for smaller stores.
	Payment gateways	Integration with secure payment gateways such as Stripe, PayPal, among others.	Transact securely using Stripe or PayPal
	Security software	SSL certificates, anti-virus software, firewalls for secure data transmission.	Add protection to the store with Sucuri or Wordfence.
Communication Requirements	Customer Notifications	SMS or email notifications for confirming orders and shipping updates. Promotion of special offers.	Enable order status notifications via Twilio for SMS, and via Mailchimp for email.
	Social Media Integrations	Links to Facebook, Instagram, etc., for marketing and engaging customers.	Sell directly with Facebook Shop and Instagram Shopping.
	APIs	Integration of the third-party API's related to shipping, customer reviews, product recommendations.	UPS API allows the automation of shipping; it can use Google Maps API to track deliveries.

Security Requirements	Data Protection	Compliance with the GDPR, CCPA, or other data protection regulations in force in your region, to guarantee safety for customer information.	Become GDPR-compliant by deploying appropriate cookie consent banners and transparent privacy policies.
	Encryption	It refers to encryption of sensitive information from end-to-end, such as customer details and payment information.	All transactions are encrypted via SSL certificates (https://).
	Fraud Detection	Is responsible for fraud prevention by offering fraud detection software to identify suspicious activities, such as suspicious login patterns or multiple attempts to make a failed payment.	With Stripe Radar or Signifyd,.

Project Development Phase Plan for Toy Station:

The Toy Station project will encompass three broad stages, which, will be characterized by distinct sets of requirements. These requirements are prioritized based on their designation thus giving priority to basic functional necessities of a system before proceeding to the other highly desirable and sophisticated systems.

Phase 1: Core Functionality Implementation :

Timeline: October 1st – October 21st
Focus: Creating the base on which Toy Station will rely on; all the fundamental elements which include user management, product management as well as the shopping cart are inherent components of the platform. These are very essential for the basic running of the system and will be the building blocks for anticipated future modifications.

Requirements Assigned to Phase 1:

1. **User Management** (Critical)
    - **Reason**: Critical in as far as the interaction of the user with the system is concerned. If user management (account creation, user login and user logout features) is not implemented, the users cannot be able to use the platform's main features.
    - **Key Features**: Sign up/Log in/Log out/Personal information and preferences changes, password retrieve.
2. **Product Management** (Critical)
    - **Reason**: Basic functions which include, browsing, searching and the capability to view products are essential to the platform. This feature is considered the center of the e-commerce function as it enables interaction between the user and the products.
    - **Key Features**: Web browsing, searching, and selecting criteria involving product filtering, sorting and returning the details view of the product.
3. **Shopping Cart and Checkout** (Critical)
    - **Reason**: Change Request CR-4098 Platform critical because it enables users to pick items and purchase them which is the primary aim of the system. The shopping cart is one of the most important components of an online store because transactions cannot occur without it due to a loss of its core function.
    - **Key Features**: Cart options include purchase, check out the contents of a cart, change quantities in a cart, delete items from a cart and check out.
4. **Backend and Frontend Integration** (Critical)
    - **Reason**: It acts as an interface between the user interface, mostly the graphical user interface (frontend) and the server (backend). This is to enable what the user does on the webpage such as adding products to cart, handle backend things like database management and requests.
    - **Key Features**: Design and use RESTful APIs for user operations including sign up, login and logout, a listing of products and shopping cart functionalities.

5. **Basic Order Management** (Critical)

    - **Reason**: It enables users to order, check status of their orders and even receive an order confirmation. They are basic expectation from the users so that they are certain that their purchase is going to be processed correctly.
    - **Key Features**: Make an order, confirm an order, and check the order status.

Phase 1 Criticality Discussion:

- **Criticality**: The functionalities that have been developed in Phase 1 are core functionalities because they define the scope and features of the Toy Station platform. If any of the characteristics do not work or are not fully developed, the platform cannot operate since people cannot engage with products or make purchases. This phase is critical to establish a healthy framework on which the other developmental phases to follow depend on.

Phase 2: Advanced Features and Enhancements:

Timeline: October 22nd – November 11th
Focus: Adding more features to the system which have the potential of improving and increasing the usage of the system by the customers, like use of recommendations, use of secure payment option and customer services among others. These features are crucial for users to stay engaged with the platform and content – thus satisfied.

Requirements Assigned to Phase 2:

1. **Recommendation Engine** (High Priority)

    - **Reason**: Recommendations are more likely to make users interested in the items they are offered since the product is chosen in accordance with a user's activity. In many ways this feature aids in improving the usability of the website, by making the shopping process easier to understand.
    - **Key Features**: If the user has a profile, smart shelves should demonstratively present the items that the user saw or bought earlier.

2. **Payment Gateway Integration** (High Priority)
    - **Reason**: Merchants' payment processing is important experience that is necessary to finalize the transaction. Secure payment interfaces are integrated because users have the confidence to purchase products, which is critical for the platform.
    - **Key Features**: Payment Gateway Integration with the third party such as Stripe, Razorpay, support multiple payment gateways, secure payment conveyance.
3. **Order Tracking, Returns, and Refunds** (High Priority)
    - **Reason**: Enables the customer to track the progress of his order, return a product, or get his money back. Such functionality adds value to the sales process and is critical for the customer, who should be able to deal with their purchase post-purchase.
    - **Key Features**: Keep track of the orders made, make and process returns, refund customers and change order status
4. **Customer Support System** (Medium Priority)
    - **Reason**: Implementing live chat and some other support services makes interactions with customers direct and effective at the same time increases customer loyalty.
    - **Key Features**: Implementation of chat support, list of frequently asked questions, and application for submitting requests.
5. **Product Reviews and Ratings** (Medium Priority)
    - **Reason**: Users believe in the opinion of others and want to be involved in a trusted network that has positive testimonials. They assist user in thinking critically on what they want to buy and the level of satisfaction they would want to have.
    - **Key Features**: Provide a form where users can write reviews about products and demonstrate the general rating for each product on the product page.

Phase 2 Criticality Discussion:

- **Criticality**: Phase 2 attributes are important for user adoption, enjoyment and platforms' effectiveness and efficiency. Though not as critical as functions outlined in Phase 1, these features are very important for the overall development of the platform as they tend to give users a sense of security and ergonomics. Payouts, which is a partial payment scheme, for the platform revenue streams are uniquely significant in this phase since they determine the generation of the platform's revenues.

Phase 3: Final Development, Optimization, and Testing:

Timeline: November 12th – December 2nd

Focus: Completion of the development of all components, achieving the best configuration of the system and valid testing. It involves readiness to make the platform ready for functional use and deployment; it also involves making the platform secure and scalable.

Requirements Assigned to Phase 3:

1. **Performance Optimization** (High Priority)
    - **Reason**: Makers of the system should make sure the system will not slow down or stagnate as more users or more transaction are processed in the system. Thus, focus on speed and performance constitutes the key to users' satisfaction.
    - **Key Features**: Backend DB queries, better distribution of loads, caching and preventing the time taken to load individual pages.
2. **Security Hardening** (High Priority)
    - **Reason**: Privacy of user information and ensuring safe payment procedures is the key factor to earning trust and platform security. The e-commerce platform security compliances are mandatory for any e-commerce platform.
    - **Key Features**: Use high-level encryption, protect API interfaces, avoid SQL injection attacks, and create access control for high risk to use at least two of the following factors: token, certificate, password.
3. **Usability and Accessibility Improvements** (Medium Priority)

- **Reason**: Makes it easy for users to interact with the platform for all they need especially for the disabled persons. This is good for getting to as many people as possible and making the atmosphere as diverse as possible.
- **Key Features**: WCAG 2.1 AA compliance should also be attained; Moreover, UI/UX shall be enhanced, and get the feedback from the users for the last touches.

4. **Full System Testing** (Critical)
    - **Reason**: Avoiding decisions that may have serious consequences or that are too risky before moving on to the next unit is it imperative to test all the components of the software.
    - **Key Features**: To execute a particular phase the unit, integration, performance, and acceptance tests should be performed.
5. **Final Deployment and Go-Live Preparation** (Critical)
    - **Reason**: It is the last stage before going live of the platform where all the set-up is finalized and the system deployed to the production setting.
    - **Key Features**: Implement server configuration, transfer the database, setting up the hosting at clouds like Amazon Web Server, and Azure, and preparing the monitoring of the live platform.

Phase 3 Criticality Discussion:

- **Criticality**: The requirements in Phase 3 are foundational to the success of the platform. All these activities guarantee that the system is optimized for performance and hardened for security at the same time while the improvements on the GUI aspect of the platform allows more users on a learning curve.

Summary of Criticality by Phase

- **Phase 1:** The critical basic components include the user management, product search, and shopping cart to enable the platform to run.
- **Phase 2:** Main priorities are payment gate and recommendations impact user's conversion while customer service and reviews increase engagement and reliability.

- **Phase 3:** A critical optimization stage and security assure adequate stability, security, and preparation for release of the platform.

Contribution Table:

Member Name	Contribution Description	Overall contribution(%)	Note
Teja Nagendra Sirigineedi	Project management and risk assessment	11%	Led risk management and project planning
Chopra Sai Arani	Project management and risk assessment	11%	Assisted in risk assessment and planning
Sai Sowjanya Edupuganti	Drafting system architecture	11%	Provided input in the form of documentation for the system architecture
Vaishnavi Shastrula	Drafting system architecture	11%	Leads the system architecture of the system
Gayathri Vutla	Functional requirements documentation	12%	Developed system functionality documentation
Preethi Medipelli	Functional requirements documentation	12%	Participated in functional requirements.
Karunya Mekala	Documenting Non-functional requirements	11%	Developed non-functional requirements
Shraehitha Reddy Banda	Documenting Non-functional requirements	11%	Helped to write the non-functional requirements
Damodar Reddy Katikala	Meeting minutes and contribution table	10%	Meetings documentation and the managed contribution table

Meeting Minutes:

Meeting 1 Minutes

- **Date**: September 10th, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet
- **Attendees**: All team members

Agenda:

1. **Introduction to Deliverable 2**:
    - Brief discussion on the essential things needed for Deliverable 2, such as the function and non-functional requirements, system architecture, and system specifications.
2. **Assigning Tasks**:
    - Each task assigned to team members
        - **Gayathri Vutla & Preethi Medipelli** :

          Functional requirements documentation.
        - **Karunya Mekala & Shraehitha Reddy Banda:**

          Non-functional requirements.
        - **Sai Sowjanya Edupuganti & Vaishnavi Shastrula:**

          Drafting system architecture.
        - **Teja Nagendra Sirigineedi & Chopra Sai Arani:**

          Overall project management, risk assessment.
        - **Damodar Reddy Katikala**:

          Meeting Minutes, Contribution table
3. **Discussion on System Architecture**:
    - Overview of the subsystems (Frontend, Backend, Database, etc.) and their interaction.
4. **Next Steps**:
    - Each team member need to start their work on assigned tasks for submission.

Next Meeting:

- **Date**: September 14th, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet

Meeting 2 Minutes

- **Date**: September 14th, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet
- **Attendees**: All team members

Agenda:

1. **Review of Progress**:
    - Team member discussed their updates on the work they had been given.
        - Functional requirements document was 50% completed.
        - The draft of the non-functional requirements was finished.
        - Working on system architecture diagram.
2. **System Architecture Discussion**:
    - Reviewed the draft architecture diagram:
        - We finalized to add more detail to the interaction between subsystems.
        - To improve clarity in the diagram, all team members gave their feedback.
3. **Plan for Requirement Finalization**:
    - We shared our thoughts to refine the functional and non-functional requirements to ensure they are aligned with project goals.
4. **Deliverable Format**:
    - Decided to submit documents in Markdown format for easy integration into GitHub.

Next Meeting:

- **Date**: September 18th, 2024
- **Time**: 4:00 PM – 5:00 PM
- **Location**: Google Meet

Meeting 3 Minutes

- **Date**: September 18th, 2024
- **Time**: 4:00 PM – 5:00 PM
- **Location**: Google Meet
- **Attendees**: All team members

Agenda:

1. **System Specifications Finalization**:
    - Functional and non-functional requirements are finalized.
    - We discussed about technical stack and tools that need to be mentioned in Deliverable 2.
2. **System Architecture Updates**:
    - Sai Sowjanya updated architecture diagram with improved subsystems interaction.
        - Payment gateway integration details has been added.
        - Flow between frontend, backend, and database subsystems has been clarified.
3. **Risk Management**:
    - Teja Nagendra Sirigineedi shared the potential risks, technical challenges and team coordination.
    - Each risk mitigations were discussed.
4. **Next Steps**:
    - Based on feedback, everyone will update their work and ready it for final review in the next meeting.

Next Meeting:

- **Date**: September 21st, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet

Meeting 4 Minutes

- **Date**: September 21st, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet
- **Attendees**: All team members

Agenda:

1. **Final Review of Documents**:
    - Functional and non-functional requirements were reviewed and finalized.
    - System architecture diagram was finalized after minor changes.
2. **Meeting Deliverable Deadlines**:
    - Each member was reminded of the upcoming submission deadlines.

- Team members shared their thoughts about potential areas of improvement to improve the deliverable.

3. **README File Preparation**:
    - Finalized the structure and content of the README.md file for the GitHub repository.
4. **Risk Management Report**:
    - Teja Nagendra shared the final risk management section, and that is approved by the team.

Next Meeting:

- **Date**: September 24th, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet

Meeting 5 Minutes

- **Date**: September 24th, 2024
- **Time**: 3:00 PM – 4:00 PM
- **Location**: Google Meet
- **Attendees**: All team members

Agenda:

1. **Final Touches for Deliverable 2**:
    - Reviewed the final documents and corrected few things related to the system requirements and architecture diagrams.
    - Updated README.md with instructions for the project structure.
2. **Submission to GitHub**:
    - All deliverables were uploaded to GitHub, including the system diagrams, requirements documentation, and meeting minutes.
3. **Final Testing and Validation**:
    - Verified that the teacher and teaching assistant could access the GitHub repository.
4. **Feedback on the Process**:
    - Each team member gave opinion on the process went till now and gave few recommendations for enhancing coordination and teamwork for future work.

Summary of Tasks Completed for Deliverable 2:

- Finalized functional and non-functional requirements.

- Completed system architecture diagrams and documentation.
- Prepared and submitted risk management reports.
- Uploaded all documents to the GitHub repository.