# SE PROJECT

## PROJECT PLAN DOCUMENT


| NAME | SRN |
| --- | --- |
| S.L.MEDHA | PES2UG21CS447 |
| SHEETAL B | PES2UG21CS493 |
| RASHMI | PES2UG21CS442 |
| SHARANYA B R | PES2UG21CS488 |


---

## 1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

We would be following the 'Scrum development model' for our project. Scrum is an agile development framework that emphasizes iterative and incremental development, allowing teams to respond quickly to changing requirements and deliver high-quality products.

In Scrum, the typical lifecycle consists of the following phases:

1. **Backlog Creation:** The product owner creates and prioritizes a backlog of features and tasks.

2. **Sprint Planning**: The team selects a set of backlog items to work on during the sprint and plans how to complete them.

3. **Sprint Execution**: The team develops the selected features during the sprint, holding daily stand-up meetings to track progress.

4. **Sprint Review**: The team demonstrates the completed work to stakeholders and receives feedback.

5. **Sprint Retrospective:** The team reflects on the sprint, identifies what went well and what could be improved, and makes adjustments for the next sprint.

To justify why this model is suitable for our project. Here are a few reasons why Scrum might be the right choice:

1**. Adaptability:**

Alignment with Changing Requirements: If your project involves evolving requirements or uncertain market conditions, Scrum's iterative nature allows you to adapt to changes quickly. The ability to reprioritize and adjust the product backlog after each sprint ensures that the project stays aligned with changing business needs.

2. **Stakeholder Collaboration:**

Regular Feedback Loops: In many projects, especially those involving software development, stakeholders often have evolving needs. Scrum's sprint review meetings provide a platform for stakeholders to see the progress made after every sprint and offer feedback. This regular feedback loop ensures that the project is always in sync with stakeholder expectations.

3. **Faster Delivery:**

Incremental Progress: Scrum divides the project into manageable chunks (sprints) where potentially shippable increments are delivered at the end of each sprint. This incremental delivery allows for the rapid release of essential features. If your project has key features that need to be deployed or tested early, Scrum facilitates this by providing a functioning product incrementally.

4. **Transparency:**

Clear Communication: Scrum ceremonies such as daily stand-up meetings, sprint planning, review, and retrospective sessions promote transparent communication within the team. Team members share progress, challenges, and plans openly, ensuring everyone is aware of the project's status. This transparency enables early identification and resolution of issues, leading to a smoother project execution.

5. **Team Empowerment:**

Cross-Functional Collaboration: Scrum emphasizes self-organizing, cross-functional teams. If your project involves diverse skill sets, Scrum allows team members to collaborate effectively, leveraging their expertise. Team members are empowered to make decisions collectively, leading to innovative problem-solving and a sense of ownership.

6. **Continuous Improvement:**

Reflective Learning: Scrum's sprint retrospective meetings encourage the team to reflect on their processes, identify what worked well, and pinpoint areas for improvement. This continuous feedback loop promotes a culture of learning and adaptation. Over the course of the project, the team can optimize its processes, leading to increased efficiency and better outcomes.

In summary, Scrum's adaptability, stakeholder collaboration, faster delivery, transparency, team empowerment, and focus on continuous improvement align with the needs of our project. By utilizing these Scrum principles and practices, we can respond effectively to changes, collaborate closely with stakeholders, deliver features incrementally, maintain clear communication, leverage diverse skills, and continuously enhance the development process. This alignment enhances the project's chances of successful execution by ensuring that it remains flexible, customer-focused, and responsive to emerging requirements and challenges.

## 2: Identify the tools that u want to use throughout the lifecycle like planning tools, design tools, version control, development tools, bug tracking, and testing tools.

## Project Management & Planning

***Tools available*:** Trello

## Design Tools

***Tools available:*** Adobe ,Sketch , Figma XD

We would be using **Figma**. It widely used by designers, developers, and product teams to create user interfaces, prototype interactions, and collaborate on design projects Figma known for its real-time collaboration features, making it popular among designers and teams working on UI/UX design projects.

## Version Control

**Tools available : Git , SVN**

We would be using **Git.** Git can handle everything from small to very large projects with speed and efficiency. Git provides robust branching and merging capabilities, allowing developers to work on multiple features concurrently and merge changes efficiently. Ensures the integrity of stored data through checksums, providing reliable version history and detection of tampering or corruption.

## Development

**Tools available : VSCode**

We would be using **VS code**( **integrated Development Environment)**

## Bug Tracking

**Tools available : Jira , Bugzilla**

We would be using **jira**. It allows extensive customization of workflows, fields, and issue types, making it adaptable to various project management needs. Jira integrates seamlessly with various development and collaboration tools, enabling smooth communication and data flow between different platforms.

## Testing Tools

**Tools available : Selenium , JUnit ,TestRail**

We would be using **selenium** . Selenium can be integrated with testing frameworks and continuous integration tools, allowing for automated testing as part of the development workflow**.**

## 3) Determine all the deliverables and categorise them as reuse/build components and justify the same.

**Reusable Components**:

**Front-End UI Components**:

- **Categorization**: Reusable
- **Justification**: Front-end UI components, such as buttons, navigation menus, and form elements, can be designed and implemented in a modular and reusable manner. They can be used consistently across various pages and can be shared among different parts of the website.

**Authentication Module**:

- **Categorization**: Reusable
- **Justification**: The authentication module, which includes user registration, login, and password reset functionality, can be designed as a reusable component. This component can be employed in other projects or for other authentication needs within the same project.

**Database Schema**:

- **Categorization**: Reusable
- **Justification**: The database schema that defines the structure of tables, relationships, and constraints can be considered a reusable component. A well-designed schema can be used as a foundation for future projects or for scaling the current system.

**APIs and Services**:

- **Categorization**: Reusable
- **Justification**: Any APIs or microservices developed for the e-commerce website, such as product catalog APIs or payment processing services, can be designed for reuse. These can be integrated into other systems or projects where similar functionality is required.

**Build Components**:

**Unique Website Design**:

- **Categorization**: Build
- **Justification**: The website's unique design, including layout, color scheme, and branding elements, is specific to the current project. While some design principles may be reused, the overall design is tailored to the website's identity.

**Eco-Friendly Product Catalog**:

- **Categorization**: Build

- **Justification**: The product catalog, which includes eco-friendly product listings and related information, is specific to the e-commerce website. It reflects the unique offerings and focus of the project and may not have direct reuse in other contexts.

**Custom Business Logic**:

- **Categorization**: Build
- **Justification**: Any custom business logic, such as algorithms for calculating carbon footprints or unique sustainability metrics, is tailored to the specific goals and requirements of the eco-friendly e-commerce website and may not be directly reusable in other projects.

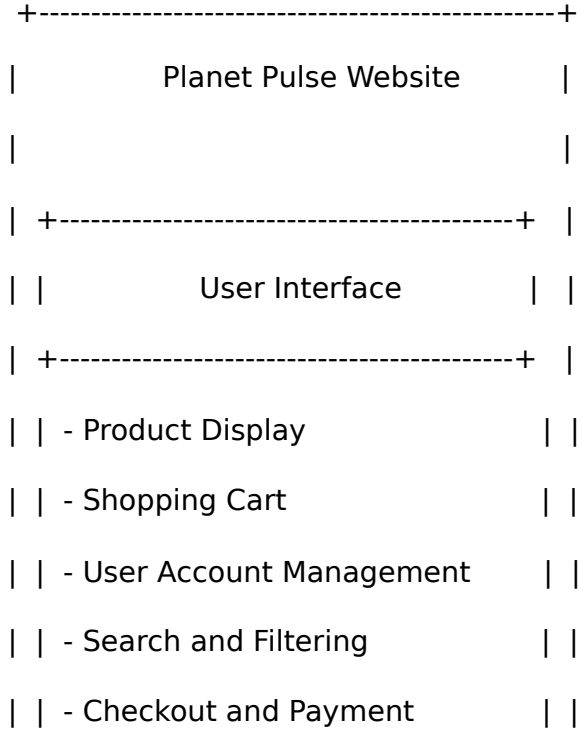
**Website Content**:

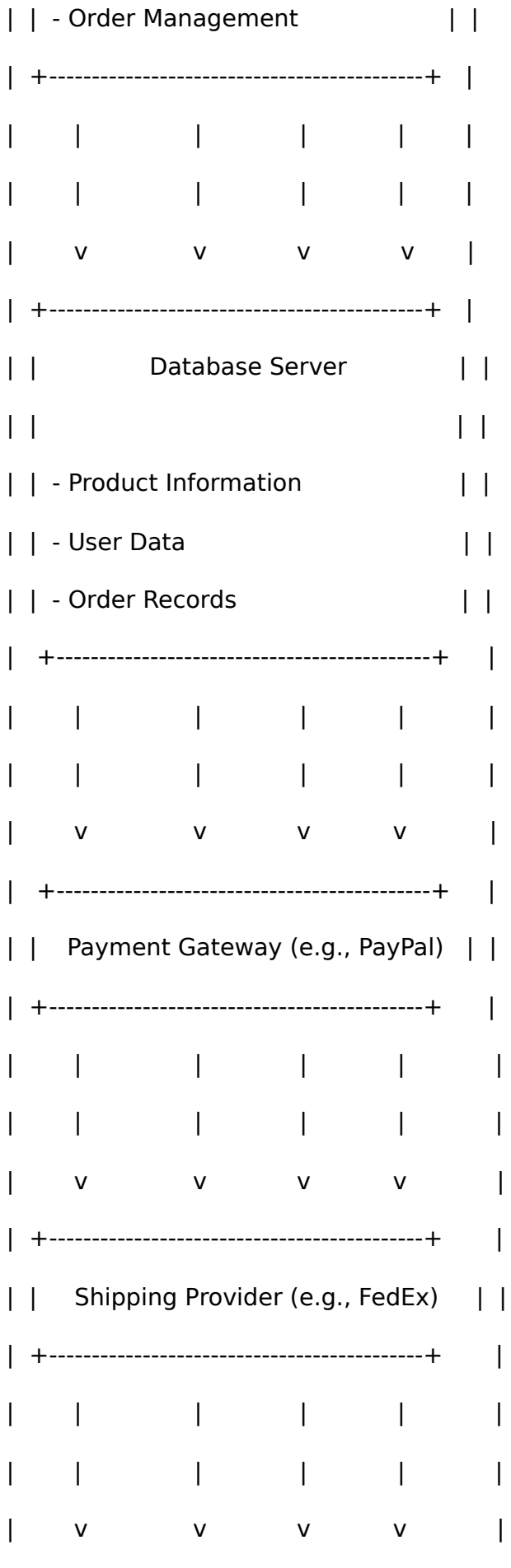
- **Categorization**: Build
- **Justification**: Content such as product descriptions, blog posts, and marketing materials is specific to the website's products and messaging. While content management systems may be reused, the content itself is unique.

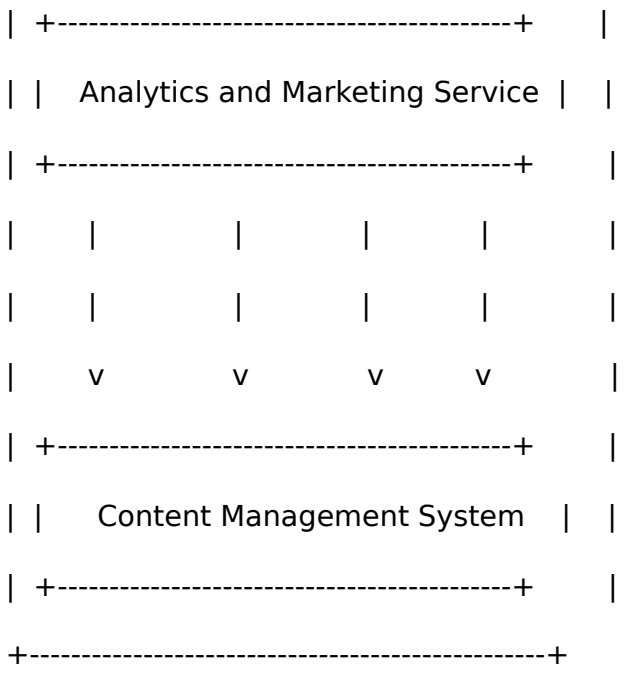
**User Experience (UX) Design**:

- **Categorization**: Build
- **Justification**: UX design elements, such as user flows and interactions, are customized for the website's specific user journey and objectives. These elements are not typically reusable as-is in other projects with different user requirements.

## 4: Create a WBS for the entire functionalities in detail.




- Order Management
Database Server
- Product Information
- User Data
- Order Records
Payment Gateway (e.g., PayPal)
Shipping Provider (e.g., FedEx)




## 5: Do a rough estimate of effort required to accomplish each task in terms of person months.

**Front-End Development**:

- Implementing the front-end UI components, including design, HTML, CSS, and JavaScript.
- **Rough Estimate**: 2 to 4 person-months depending on the complexity of the design and interactivity.

**Authentication Module**:

- Developing user registration, login, and password reset functionality.
- **Rough Estimate**: 1 to 2 person-months depending on the complexity of security features and integration.

**Database Design and Implementation**:

- Designing the database schema, creating tables, defining relationships, and implementing CRUD (Create, Read, Update, Delete) operations.
- **Rough Estimate**: 2 to 3 person-months, considering the complexity of the schema and database optimization.

**APIs and Services**:

- Developing APIs for product catalog, payment processing, and other core functionalities.
- **Rough Estimate**: 2 to 4 person-months depending on the number and complexity of APIs.

**Unique Website Design**:

- Designing the website's unique layout, color scheme, and branding elements.

- **Rough Estimate**: 2 to 4 person-months, depending on the complexity of design requirements.

**Eco-Friendly Product Catalog**:

- Implementing the product catalog, including features like product listings, search, and filtering.
- **Rough Estimate**: 2 to 3 person-months, considering the number of products and complexity of features.

**Custom Business Logic**:

- Developing custom algorithms for calculating carbon footprints or sustainability metrics.
- **Rough Estimate**: 1 to 2 person-months, depending on the complexity of the calculations.

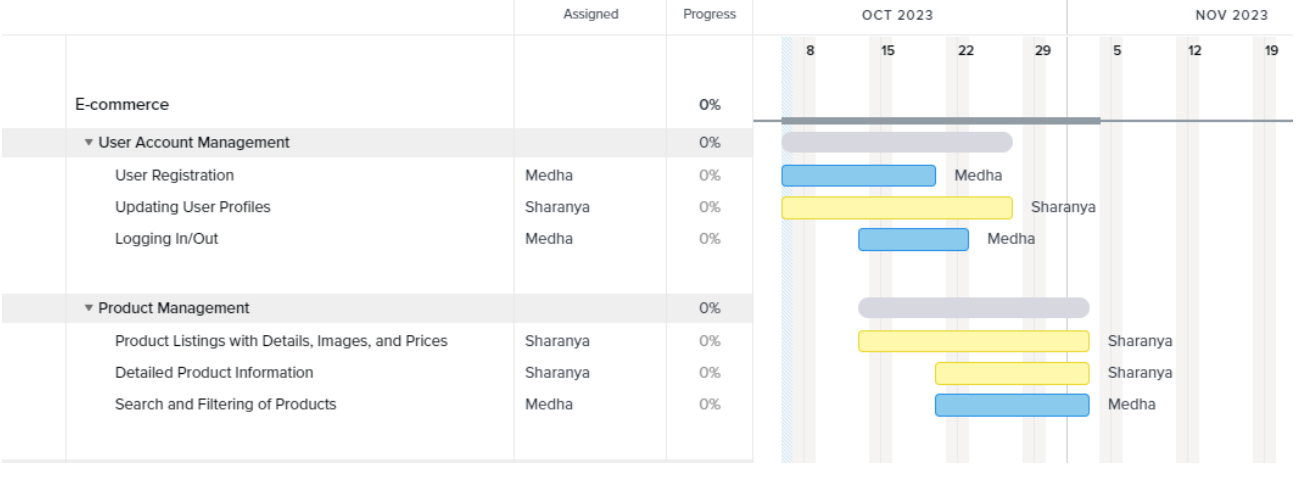
**Content Creation and Management**:

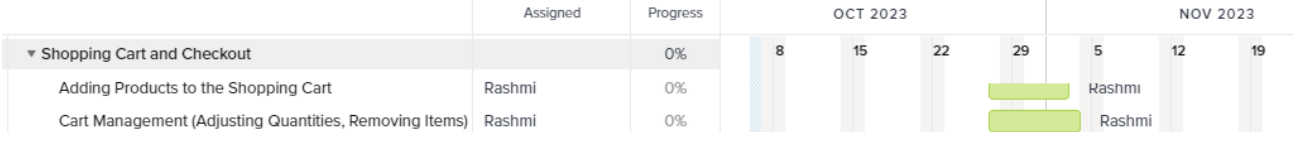
- Generating website content, including product descriptions and blog posts.
- **Rough Estimate**: Ongoing effort, but initially 1 to 2 person-months to create the initial content.

**User Experience (UX) Design**:

- Designing user flows, interactions, and wireframes.
- **Rough Estimate**: 2 to 3 person-months, depending on the complexity of UX requirements.

## 6: Create the Gantt Chart for scheduling using any tool.



| | | | |
|---|---|---|---|
| Proceeding Through the Checkout Process | Rashmi | 0% | Rashmi |
| Selecting Shipping Options | Rashmi | 0% | Rashmi |
| Securely Entering Payment Information | Medha, Rashmi | 0% | Medha, Rashmi |
| ▼ Order Management | | 0% | |
| Viewing Order History | Sheetal | 0% | Sheetal |
| Checking Order Status | Sheetal | 0% | Sheetal |
| Tracking Orders and Estimated Delivery Dates | Sharanya, Sheetal | 0% | Sharanya, Sheetal |
| Managing Orders, Updating Statuses, and Tracking Deli... | Sheetal | 0% | Sheetal |