**SW-PM SE Project (Phase I)**

**\_\_\_\_\_\_\_\_**

**15**

|  |  |
| --- | --- |
| Project Name | Task Tracking Tool |
| Online Website address | N/A |
| GitHub Link | https://github.com/Mohamedbasem1/SPM-Project |
| Jira link | https://mohamedbasem.atlassian.net/jira/software/projects/KAN/boards/1?atlOrigin=eyJpIjoiODFhOGZlZTA2OWJkNDE3ZDlmZTc2YWE3OGI1NWRjYzYiLCJwIjoiaiJ9 |

1. **Team Members:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Signature | Grade |
| 224179 | Islam Abdulhakeem |  |  |
| 220017 | Mohamed Basem |  |  |
|  | Mohamed Mabrouk |  |  |
|  | Hisham Ahmed |  |  |
|  | Ahmed Hisham |  |  |

1. **Grading Criteria:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | | **Actual Grade** | **Notes** |
| ***Documentation*** | ***10*** |  |  |
| Detailed Project Plan | **3** |  |
| Project Structure and process details | **2** |  |
| Sprints Planning | **1** |  |
| Non-technical tasks planning | **2** |  |
| Tasks breakdown with details | **2** |  |
| ***Application Planning*** | ***5*** |  |  |
| Implementation system Architecture | **1** |  |
| Configuration management and version control Planning | **1** |  |
| Performance KPIs identification | **1** |  |
| CRUD screens prototyping and system workflow | **2** |  |

N.B. Deliverables of the project plan should be covering items in the project plan template starting from project overview to the project budgeting

**Instructor Signature:**

**Detailed Project Plan:**

**1. Project Initiation Phase:**

**1.1 Define Project Scope and Objectives:**

 Create a simple, easy-to-use task tracking tool.

 Add helpful features like organizing tasks, tracking progress, and working with teammates.

 Finish and release the tool in six months.

**1.2 Conduct Market Research:**

 Find out who will use the tool, like teams, freelancers, or businesses.

 Learn how people currently manage their tasks and what they wish was better.

 Look at other tools to see what they’re good at and where they fall short.

 Make a list of things that make our tool special, like being super easy to use or having great collaboration features.

**1.3 Create Project Charter:**

Project Purpose:

To help people and teams stay on top of their work, finish tasks on time, and work better together.

Project Objectives:

 Make a tool that’s easy to use for planning and tracking tasks.

 Add useful features like deadlines, progress updates, and team chats.

 Get everything done on time and within the budget.

Stakeholders:

** Sponsor**.

 **Project Manager**.

 **Development Team**.

** Marketing Team**.

 **Users.**

Success Criteria:

 People like the tool and say it’s helpful.

 A lot of people start using it in the first six months.

 We launch it on time and stay within budget

Risks:

 Delays if something technical doesn’t work as expected.

 Challenges in making the tool work smoothly for both individuals and teams.

 Other tools might already have a strong user base.  
  
**2. Planning Phase:**

**2.1 Develop Work Breakdown Structure (WBS):**

**Task Breakdown:**

* Identify main tasks: Frontend Design, Backend Development, Task Management Features, Collaboration Tools, Testing, and Launch.
* Split these main tasks into smaller, more specific tasks for better planning.

**WBS Documentation:**

Create a clear plan that lists all tasks, subtasks, and how they connect to each other.

**2.2 Create Gantt Chart:**

**Task Schedule:**

* Use the WBS to make a Gantt chart showing timelines, important milestones, and task dependencies.

**Assign Work:**

* Match tasks to team members based on their skills.
* Make sure everyone has a fair workload and resources.
  1. **Resource Planning:**

**Identify Team Members:**

* List all team members and their roles in the project.
* Highlight any special skills needed for specific tasks (e.g., UI/UX design or database setup).

**Assign Roles:**

* Assign team members to tasks based on what they’re good at and when they’re available.
* Set up clear communication methods to keep everyone on the same page.

**2.4 Risk Management:**

**Identify Risks:**

* Conduct a thorough risk assessment to identify potential issues.
* Categorize risks as technical, operational, or external.

**Risk Mitigation Strategies:**

* Develop strategies to mitigate each identified risk.
* Assign responsibility for risk monitoring and mitigation.

**3. Design and Development Phase:**

**3.1 Design Website Layout:**

**Wireframes and Visual Elements:**

 Work with UI/UX designers to draw rough layouts (wireframes) showing how the tool will look.

 Design buttons, menus, and screens that are visually pleasing and simple to navigate.

 Ask for feedback, fix anything confusing, and repeat until it's awesome.

**3.2 Develop task tracking tool Platform:**

**Backend Infrastructure:**

* Set up the backend infrastructure using Node.js with Express.
* Create APIs for communication between frontend and backend.

**Add Features:**

* Enable users to create, edit, and organize tasks.
* Add filters, deadlines, and categories to make life easier.
* Build a team collaboration system where users can assign tasks and track progress.

**3.3 Make Sure Everyone Can Log In Safely:**

* **Login and Register:**
  + Build a simple and secure system for users to sign up and log in.
  + Add "forgot password" options because, let’s face it, people forget things.
  + For the tech-savvy, include social login options (like Google or LinkedIn).
* **Who Gets to Do What:**
  + Set up roles: regular users for tracking tasks and admins to oversee everything.
  + Make sure permissions are tight so no one sees what they shouldn’t.

**3.4 Build the Admin Area (The Control Center):**

* **For the Bosses:**
  + Create a special dashboard where admins can manage users and their roles.
  + Add tools to check tool performance and user activity.
  + Give admins the ability to tweak things as needed.

**3.5 Connect the Front and Back (Frontend):**

* **Put It All Together:**
  + Make the frontend pretty, fast, and easy to use.
  + Ensure buttons actually do what they’re supposed to (no clicking frustration).
  + Test it with real users—if something breaks, fix it ASAP.

**4. Testing Phase:**

**4.1 System Testing:**

**Test the Features:**

* Check every part of the tool—can users create tasks, update them, track progress, and collaborate smoothly?
* Make sure everything works the way it’s supposed to, with no surprises.

**Check Speed and Performance:**

* Test the tool under different conditions, like when a lot of people are using it at once.
* Fix anything that slows it down so it’s fast and responsive.

**Security Assessment :**

* Test for security holes—better we find them than someone else!
* Simulate possible attacks to make sure the tool is safe and sound.

**4.2 User Acceptance Testing (UAT):**

**Stakeholder Engagement:**

* Invite project sponsors, team members, and actual users to test the tool in real-world scenarios.
* Give them simple instructions and let them explore how it works.

**Listen to Their Thoughts:**

* Gather feedback on what they like, what confuses them, and what could be better.
* Use their suggestions to improve the tool and make it more user-friendly.

**4.3 Issue Resolution:**

**Issue Identification:**

 Review all the test results and note down any bugs or glitches.

 Prioritize the most serious issues first, especially ones that could frustrate users.

**Fix and Test Again:**

 Work with the development team to squash the bugs.

 After fixing, test everything again to make sure no new problems pop up.

**Iterative Testing:**

 Repeat the process as needed until the tool is polished and meets all standards.

 Keep stakeholders in the loop about what’s been fixed and what’s still being worked on.

**5. Deployment Phase:**

**5.1 Launch the Task Tracking Tool:**

**Public Release:**

 Make the tool available for everyone to use.

 Double-check that all features work and the interface is smooth and easy to navigate.

**Performance Monitoring and Maintenance:**

* Implement monitoring tools to track post-deployment performance and stability.
* Set up regular maintenance to fix any unexpected issues quickly.

**5.2 Spread the Market:**

* **Run Campaigns:**
  + Start marketing campaigns to let people know about the tool.
  + Use social media, email, and online ads to reach as many users as possible.
* **Offer Deals:**
  + Use promotions like discounts for early adopters or free trials.
  + Partner with influencers or businesses to get more visibility.

**5.3 Set Up User Support:**

* **Help When They Need It:**
  + Provide support through email, chat, or phone so users can get help easily.
  + Make it clear how they can contact support or report issues.
* **Use Smart Tools:**
  + Set up tools to track user problems and resolve them quickly.
  + Train the support team to handle all kinds of questions and issues.
* **Create Guides:**
  + Write clear instructions and FAQs to answer common questions.
  + Make sure users can find these guides easily on the tool’s website.

**6. Monitoring and Maintenance:**

**6.1 Continuous Performance Monitoring:**

 **Watch How It’s Running:**

* Use monitoring tools to check how well the tool is performing, like how fast it loads or handles tasks.

 **Listen to Users:**

* Collect feedback from users about what’s working and what’s not.
* Use their comments to find areas for improvement.

 **Catch Problems Early:**

* Spot potential issues before they become big problems by keeping an eye on performance data.
* Fix the most important issues first to keep users happy and the tool stable.

**6.2 Regular Updates and Fixes:**

* **Update the Tool:**
  + Set a schedule to regularly update the software with bug fixes, security improvements, and new features.
* **Tidy Up the Database:**
  + Check the database to make sure it’s running smoothly and update queries to keep things fast.
* **Keep It Secure:**
  + Run security checks often to find and fix any vulnerabilities.
  + Stay informed about new security risks and prepare defenses.
* **Review Content:**
  + Update any outdated information, guides, or help articles so users always have accurate info.
* **Backup Everything:**
  + Make sure backups of the tool’s data are done regularly and can be restored quickly if needed.
* **Make It Faster:**
  + Use what you learn from monitoring and user feedback to optimize things like images and scripts for a better, faster experience.

**7. Project Closure:**

**7.1 Finalizing Documentation:**

* **Organize Everything:**
  + Collect all project documents, like plans, requirements, and progress reports, and organize them neatly for future reference.
* **Write the Final Report:**
  + Summarize the project in one clear report—what worked, what didn’t, and lessons we learned along the way.

**7.2 Reflect on the Journey:**

* **Hold a Wrap-Up Meeting:**
  + Get the team and stakeholders together to celebrate the successes and discuss the milestones we hit.
* **Learn from Challenges:**
  + Talk about the hurdles we faced during the project. What can we do better next time?

**7.3 Pass It to the Operations Team:**

* **Plan the Handoff:**
  + Create a clear plan for transitioning the tool to the team who will manage it daily.
  + Ensure roles and responsibilities are well-defined, and support channels are in place.
* **Train the Team:**
  + Teach the operations team how to use and maintain the tool confidently. Answer their questions and guide them through the features.
* **Smooth Handover:**
  + Ensure the project transitions seamlessly, with no loose ends. Share all resources they’ll need for ongoing success.

**7.4 Officially Close the Project:**

* **Document the Finish:**
  + Write an official document marking the project as complete and get approval from stakeholders.
  + Celebrate the successful delivery of the **Task Tracking Tool!**

**Key Milestones:**

* Project Kickoff
* Tool Launch
* Completion of User Testing
* Handover to Operations

**Tools and Technologies Used:**

* **Project Management:** Jira
* **Design Tools:** Figma or Adobe XD
* **Development Tools:** Node.js, React, or similar

**Communication Plan:**

* Regular project update meetings.
* Clear communication channels for team collaboration.
* Stakeholder meetings to keep everyone in the loop.

**Project Structure:**

**1. Project Team:**

**1.1 Project Manager:**

Roles**:**

* Oversee and manage the entire project.
* Stay in touch with stakeholders and keep everyone updated.
* Distribute resources and assign tasks to the team.

**1.2 Web Development Team:**

Responsibilities**:**

 Work on both the frontend and backend of the tool.

 Add features for task tracking and collaboration.

 Ensure smooth integration of necessary tools and systems.

**1.3 UI/UX Designers:**

Responsibilities:

 Design wireframes and user-friendly interfaces.

 Create the layout and visual elements of the tool.

 Make sure it’s easy to navigate and enjoyable for users.

**1.4 Quality Assurance (QA) Team:**

Responsibilities:

 Test the system to ensure everything works properly.

 Find and report bugs.

 Coordinate user testing to make sure it meets the needs of users.

**1.5 Marketing Specialist:**

Responsibilities:

 Develop strategies to promote the tool.

 Run campaigns to spread the word and attract users.

**1.6 Customer Support Representatives:**

Responsibilities:

 Set up processes for user support.

 Help users with any issues or questions they have.

 Ensure users are happy with the tool.

**2. Development Environments:**

**2.1 Development Server:**

* Used for ongoing development and testing.
* Hosts the latest version of the application.

**2.2 Staging Server:**

 A test environment that simulates the live tool.

 Used for final testing before launch.

**2.3 Production Server:**

 The live version of the tool, accessible to end users.

 Hosts the final, fully tested version.

**3. Tools and Technologies:**

**3.1 Version Control:**

* **Git** for managing code versions and collaboration.

**3.2 Project Management Software:**

* **Jira** or similar tools to keep track of tasks and progress.

**3.3 Design Tools:**

* **Figma** or **Adobe XD** for designing the user interface.

**3.4 Development Framework:**

* **Use suitable frameworks or platforms like** Node.js**,** React**, or others for building the tool.**

**Process Details:**

**1. Development Process:**

**1.1 Agile Development:**

* Use **Agile** (like **Scrum**) to break the project into manageable chunks, called sprints.
* Work in short cycles to improve and deliver features incrementally.

**1.2 Collaborative Development:**

* Hold regular meetings (stand-ups) to discuss progress, challenges, and next steps.

**1.3 Code Review:**

Regular code review sessions to ensure quality.

**2. Design Process:**

**2.1 Wireframing and Prototyping:**

Develop wireframes for website layout.

Create interactive prototypes for user testing.

**2.2 UI/UX Design:**

Continuously improve the design based on user and stakeholder feedback.

**2.3 Feedback and Approval:**

Gather feedback from stakeholders.

Get final design approval before development begins.

**3. Testing Process:**

**3.1 System Testing:**

 Test the entire tool to make sure everything works as it should.

 Identify and fix any bugs or issues.

* 1. **User Acceptance Testing (UAT):**
* Stakeholders and end-users test the system.
* Make final adjustments based on their feedback.

**4. Deployment Process:**

**4.1 Deploy to Staging:**

 Deploy the tool to a staging environment to simulate the live system.

 Perform final testing to make sure everything works.

**4.2 Production Deployment:** Deploy the fully-tested tool to the live server for end-users.

 Monitor the system for any issues after it goes live.  
  
**5. Marketing and Customer Acquisition:**

**5.1 Marketing Strategies:**

Launch planned marketing campaigns using digital channels to attract users.

**5.2 Customer Acquisition:**

 Track how many new users sign up and how effective the marketing is.

 Adjust strategies based on performance.  
  
**6. Customer Support:**

**6.1 Support Processes:**

 Set up processes to help users with any questions or issues.

 Train the support team on how to handle customer queries.

**6.2 Issue Resolution:**

* Resolve customer queries and issues promptly.
* Collect feedback for continuous improvement.

**7. Monitoring and Maintenance:**

**7.1 Performance Monitoring:**

 Use tools to keep track of how well the tool is performing.

 Regularly check the health of the server and ensure the system is responsive.**7.2 Regular Updates:**

 Plan and schedule regular updates to improve security and add new features.

 Make sure updates don’t disrupt the live tool for users.

**8. Project Closure:**

**8.1 Documentation:**

* Compile project documentation, including lessons learned.
* Archive all relevant project files and assets.

**8.2 Handover to Operations:**

 Hand over the project to the teams who will manage it day-to-day.

 Provide training and the necessary documentation to ensure smooth ongoing operation.

**Sprint Planning for Task Tracking Tool Development**

**Sprint Details:**

* **Sprint Duration:** 3 weeks
* **Sprint Start Date:** 02/15/2024
* **Sprint End Date:** 03/08/2024

**Goals for the Sprint:**

* **Primary Goal:** Make the tool mobile-friendly with responsive design.
* **Secondary Goals:**
  + Add another payment method.
  + Improve task search functionality.
  + Start working on user review feature.
  + Run initial security tests.

**Non-Technical Tasks Planning**

**1. Marketing and Promotion:**

* **Develop Marketing Strategy:** Research the target audience and plan how to promote the tool.
* **Create Promotional Materials:** Design online and offline materials.
* **Implement Digital Marketing Campaigns:** Run social media and email campaigns.
* **Partnership Outreach:** Find partners for cross-promotion.

**2. Customer Support Setup:**

**2.1 Define Support Processes:**

Develop standard operating procedures for handling customer inquiries and issues.

**2.2 Training for Support Representatives:**

Provide training to support representatives on product knowledge and communication skills.

**2.3 FAQs and Knowledge Base:**

Develop and maintain a Frequently Asked Questions (FAQs) section and knowledge base.

**3. Legal and Compliance:**

 **Terms and Conditions:** Write clear terms for users to agree to.

 **Privacy Policy:** Create a privacy policy to protect user data.

 **Payment Security Compliance:** Ensure secure transactions and follow industry standards.

 **Legal Consultation:** Get legal advice to make sure everything is compliant.

**4. Project Communication:**

**4.1 Stakeholder Communication Plan:**

Define a stakeholder communication plan for regular updates and progress reports.

**4.2 Internal Team Communication:**

Set up effective communication channels for the project team.

**4.3 Customer Communication Channels:**

Establish ways for customers to provide feedback or ask questions.

**5. Documentation and Reporting:**

 **Project Documentation:** Keep track of all project plans, notes, and updates.

 **Performance Reporting:** Regularly report on progress and tool performance.

 **Final Project Report:** Write a final report with lessons learned and project outcomes.  
  
**Tasks Breakdown**

**Project Initiation:**

**1.1 Define Project Scope:**

Gather requirements from stakeholders to understand project needs.

Draft a project scope document detailing project boundaries, objectives, and deliverables.

**1.2 Market Research:**

Conduct a thorough analysis of the target market and competitors.

**1.3 Create Project Charter:**

Outline project goals, milestones, and key stakeholders.

Finalize the project charter with stakeholder approval.

**Planning:**

 **Work Breakdown:** List major deliverables and break tasks down into smaller steps.

 **Gantt Chart:** Map out tasks on a timeline and update as needed.

 **Resource Allocation:** Assign team members based on skills and availability.

 **Risk Management:** Identify potential risks and develop plans to minimize them.

**Design and Development:**

 **Design Tool Interface:** Create wireframes and design layouts.

 **Payment Integration:** Add a secure payment system.

 **Product Management:** Implement features for adding and updating tasks and categories.

 **User Authentication:** Set up secure login and account recovery options.

**Testing:**

**4.1 System Testing:**

Test all functionalities and ensure everything works as expected.

**4.2 User Acceptance Testing:**

Invite stakeholders and end-users to test the system.

Address and promptly resolve any issues identified during testing.

**Deployment:**

 **Deploy Task Tracking Tool:** Move the tool to a live environment.

 **Marketing:** Launch marketing campaigns and monitor their effectiveness.on result.

* **Customer Support Setup:**

-Train customer support representatives on the online store's features and common issues.

-Set up effective customer support channels, including chat, email, and phone support.

**Monitoring and Maintenance:**

**6.1 Monitor Performance:**

Implement tools for ongoing performance monitoring, including website responsiveness and server health.

Regularly review performance metrics to identify and address any issues promptly.

**6.2 Regular Updates:**

Schedule regular updates for bug fixes, security patches, and feature improvements.

Communicate updates to users through appropriate channels to ensure transparency.

**Project Closure:**

 **Documentation:** Collect all documents and finalize the project report.

 **Review and Evaluation:** Get feedback from the team and stakeholders to improve future projects.

 **Handover:** Make sure all tasks are completed and hand over the project to operational teams with necessary training and documentation.

### ****Implementation System Architecture: Task Tracking Tool****

#### **1. Frontend (What the Users See)**

* **User Interface (UI):**
  + Design web pages where users can view tasks, create new tasks, and manage their workflow.
  + Use **HTML**, **CSS**, and **JavaScript** to create an intuitive and user-friendly design.
* **Client-Side Framework:**
  + Leverage frameworks like **React** or **Vue.js** for smooth, dynamic interactions and responsiveness.
* **User Authentication:**
  + Implement secure login and registration features to manage user access.
  + Enable password recovery and multi-factor authentication options.

#### **2. Backend**

* **2.1 Server:**
  + Use a server-side framework like **Node.js** or **Django** to handle user requests and system responses.
  + Develop **RESTful APIs** for communication between the frontend and backend systems.
* **2.2 Database:**
  + Choose a reliable database system (e.g., **MySQL** or **PostgreSQL**) to store task details, user information, and activity logs.
  + Design optimized schemas to support efficient task retrieval and updates.
* **2.3 Business Logic:**
  + Develop processes for creating, updating, and deleting tasks.
  + Implement features like task prioritization, deadlines, and status updates.

#### **3. Task Management Features**

* **3.1 Task Creation and Management:**
  + Provide an interface for creating, editing, and deleting tasks.
  + Include support for task categories, tags, and attachments.
* **3.2 Workflow Management:**
  + Enable users to assign tasks to team members and monitor their progress.
  + Integrate functionalities for setting deadlines and reminders.
* **3.3 Collaboration:**
  + Allow comments and discussions within tasks to facilitate teamwork.
  + Include real-time notifications for updates.

#### **4. Notifications System**

* **4.1 Task Notifications:**
  + Implement email and in-app notifications for task updates and reminders.
* **4.2 Event Triggers:**
  + Notify users of upcoming deadlines, completed tasks, or changes in task status.

#### **5. User Authentication and Authorization**

* **5.1 Secure Login:**
  + Implement robust user login and account recovery processes.
  + Use encryption to protect sensitive user information.
* **5.2 Access Control:**
  + Define roles (e.g., Admin, Team Member) and permissions for task visibility and actions.

#### **6. Security Measures**

* **6.1 SSL/TLS Encryption:**
  + Enforce SSL/TLS encryption for all data exchanges.
* **6.2 Data Validation:**
  + Validate user inputs to prevent vulnerabilities like **SQL injection** and **cross-site scripting (XSS)** attacks.

#### **7. Monitoring and Analytics**

* **7.1 Performance Monitoring:**
  + Implement tools (e.g., **New Relic**, **Datadog**) to monitor backend performance and application uptime.
* **7.2 Analytics Integration:**
  + Use analytics tools (e.g., **Google Analytics**) to track user activity and gain insights into task management trends.

**Configuration Management and Version Control Planning**

**1. Version Control System (VCS):**

**1.1 Selection:**

The online task tracking tool project will utilize Git as the chosen version control system. Git is selected for its powerful, it’s popular, and honestly, it just works like magic.

**1.2 Repository Setup:**

Establish a central Git repository hosted on GitHub, providing a collaborative platform for version control.

Adhere to the feature branching strategy, allowing developers to work on isolated features and improvements.

**1.3 Collaboration Guidelines:**

Developers are expected to clone the repository and create feature branches for their tasks.

Pull requests are mandatory for code review and should include detailed commit messages.

Commits should be frequent and focused, contributing to better traceability and collaboration.

**2. Branching Strategy:**

**2.1 Feature Branches:**

Developers create feature branches for the implementation of new features or enhancements.

Feature branches undergo thorough code reviews before being merged into the main branch.

**2.2 Release Branches:**

Release branches are established for stable versions to facilitate testing and deployment preparations.

Release branches merge into both the main branch and long-term support branches.

**2.3 Hotfix Branches:**

Hotfix branches are created to promptly address critical issues in the production environment.

Hotfix branches are merged into both release and main branches to ensure consistency.

**3. Change Management:**

**3.1 Change Requests:**

* All changes, whether enhancements or bug fixes, must be proposed through documented change requests.
* A designated change control board reviews and approves change requests before implementation.

**3.2 Code Reviews:**

 Every code change gets a second pair of eyes—think of it as proofreading for your code.

 Reviewers check if it follows the rules, works as expected, and can’t secretly break things.

**3.3 Approval Workflow:**

 No lone wolves here! All changes need approval from the **Project Manager** and key team members.

 Use **pull requests** to discuss, review, and get the green light for your changes.

**4. Build and Deployment Automation:**

 **4.1 Continuous Integration (CI):**

* **Jenkins** is our build buddy! It handles the boring stuff, like compiling code every time you push to the main branch.

 **4.2 Automated Testing:**

* Before anything goes live, tests (unit, integration, and end-to-end) check if your code behaves. If tests fail, no deployment—it’s like a safety lock!

 **4.3 Deployment Scripts:**

* Scripts handle deployments, so you don’t have to. Just run them, and voilà!
* Everything goes to a **staging environment** first, like a dress rehearsal before the big show.

**5. Documentation Management:**

**5.1 Documentation Repositories:**

Maintain separate repositories for project documentation, including technical documentation, user guides, and system manuals.

Utilize the same version control system (Git) for documentation to track changes over time.

**5.2 Versioned Documentation:**

Ensure that documentation is versioned in sync with the codebase.

Include release notes and changelogs in documentation repositories for clear communication.

**6. Release Planning:**

**6.1 Versioning Scheme:**

* Adopt Semantic Versioning (SemVer) for versioning releases.

 Make sure everyone knows how it works, so we’re all on the same page.

**6.2 Release Schedule:**

 Plan releases when features are ready and bugs are squashed.

 Align with business goals (and maybe even a marketing push—nothing like a grand launch).  
  
 **Performance KPIs Identification for Online Store Project**

**1. Website Load Time:**

* **KPI: Average page load time**
* **Objective: Achieve an average page load time of under 3 seconds to ensure a seamless and responsive user experience.**
* **Measurement Tools: Google Page Speed Insights, Webpage Test**
* **Target: <3 seconds**

**2. Server Response Time:**

* **KPI: Average server response time**
* **Objective: Maintain an average server response time below 200 milliseconds for quick interaction between users and the server.**
* **Measurement Tools: Application performance monitoring (APM) tools, server logs**
* **Target: <200 milliseconds**

**User Engagement Magic:**

* **KPI:** Conversion rate (e.g., task completions or goal tracking)
* **Objective:** Get users to complete **5% more tasks**—more goals achieved means happy users.
* **Tools:** User analytics, internal tracking systems
* **Target:** **>5%**

**Smooth Transactions:**

* **KPI:** Time to process tasks or requests
* **Objective:** Make sure every action is processed in under **2 seconds**—because waiting is so last decade!
* **Tools:** Monitoring tools, logs
* **Target:** **<2 seconds**

**5. Error Rate:**

* **KPI: Percentage of transactions or page views resulting in errors**
* **Objective: Maintain an error rate below 0.5% to ensure a smooth and error-free user journey.**
* **Measurement Tools: Error tracking tools, logs, user feedback**
* **Target: <0.5%**

**6. Always Available:**

* **KPI: Percentage of time the website is available and operational**
* **Objective: Ensure a website uptime of 99.9% to prevent disruptions in user access.**
* **Measurement Tools: Website monitoring services, uptime tracking tools**
* **Target: >99.9%**

**7. Peak Traffic Handling:**

* **KPI: Ability to handle peak traffic without performance degradation**
* **Objective: Ensure the website can handle a 20% increase in traffic during peak hours without degradation.**
* **Measurement Tools: Load testing tools, performance monitoring during peak hours**
* **Target: 20% increase without degradation**

**8. Mobile-First Focus:**

* **KPI:** Mobile performance
* **Objective:** Be lightning-fast on mobile devices with load times under **3 seconds**.
* **Tools:** Mobile performance tools
* **Target:** **<3 seconds**

**9. Database Performance:**

 **KPI:** Database response time

 **Objective:** Answer database queries in less than **100 milliseconds**—it’s all about efficiency.

 **Tools:** Database monitoring tools

 **Target:** **<100 ms**

**10. User Insights:**

* **KPI:** Customer journey tracking
* **Objective:** Map out user behavior to improve how tasks flow and help users stay on track.
* **Tools:** Behaviour tracking tools

**11. Scalability:**

* **KPI: Ability to scale infrastructure based on demand**
* **Objective: Ensure the system can scale efficiently to handle a 50% increase in user base.**
* **Measurement Tools: Load testing tools, cloud infrastructure monitoring**
* **Target: 50% increase without performance degradation**

**12. Search Performance:**

 **KPI:** Search speed and accuracy

 **Objective:** Let users find what they’re looking for in under **1 second**—quick, smart, and spot-on!

 **Tools:** Search analytics, response time monitoring

 **Target:** **<1 second**

**CRUD Screen Prototyping System Workflow:**

A screenshot of a computer

Description automatically generated  
  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated