Pain Point identified by me:

The user interface in Notion is intimidating and overwhelming, with too many templates and features that make users spend more time figuring out how to work rather than actually working.The user learning time should be reduced.

1. **Consumer Needs:**
    - A more intuitive and user-friendly interface.
    - Progressive learning and user levelling system (beginner, intermediate, advanced).

2. **Product Goals:**
    - Implement a levelled system that adapts to the user's proficiency.
    - Improve overall user satisfaction and decrease the time spent on learning the tool.

3. **User Stories:**
    - As a new user, I want a guided onboarding process so that I can quickly understand how to use the basic features of Notion.
    - As a user, I want a simplified navigation system so that I can easily find the tools and features I need.
    - As a beginner, I want to start with a simplified interface and gradually unlock more features as I become more proficient.

4. **Testing Strategy:**

Test Plan:

Objective:

- Ensure the new UI design meets the goal of being more intuitive and reducing the learning time for users.
- Verify that all functionalities work as intended without introducing new bugs.

Scope:

- **Functional Testing:** Verify that all UI elements function correctly.
- **Usability Testing:** Ensure the new UI is easy to use and understand.
- **Performance Testing:** Ensure the new UI does not negatively impact the performance of the application.

Resources:

- **Roles:** Assign roles for test planning, execution, and analysis (e.g., QA Lead, Test Engineers, UX Researchers).
- **Tools:** Use Notion for task management and documentation, and tools like UserTesting for usability tests and JMeter for performance testing.

Schedule:

- **Timeline:** Create a timeline with deadlines for each phase of testing.

Test Cases:

Functional Testing:

- **Scenario 1:** Verify that the new navigation menu works as expected.
  - **Steps:** Click each menu item and verify it navigates to the correct page.
  - **Expected Result:** Each click leads to the appropriate page without errors.
- **Scenario 2:** Check that all buttons are responsive and trigger the correct actions.
  - **Steps:** Click all buttons on various pages and verify their actions.
  - **Expected Result:** Each button performs its designated action correctly.

Usability Testing:

- **Scenario :** Test ease of navigation for new users.

    - **Steps:** Users perform common tasks.
    - **Expected Result:** Users should be able to complete tasks without confusion.

Performance Testing:

- **Scenario 1:** Measure load times for pages with the new UI.
- **Scenario 2:** Test the responsiveness of the UI under different conditions.

Quality Assurance:

Initial Build Quality:

- Conduct thorough code reviews to ensure code quality and adherence to design principles.
- Implement automated tests to catch issues early.
- Perform regular regression tests to ensure new updates do not introduce new bugs.
- Collect and analyse user feedback continuously to identify areas for improvement.

Post-Launch Updates:

- Monitor user interactions and performance metrics post-launch to identify any issues.
- Quickly address any post-launch bugs or performance issues with timely patches and updates.
- Use feedback and performance data to iteratively improve the UI.

5. Roadmap:

Phase 1: Concept and Design

Research: Conduct user research to identify pain points and gather insights.

- **Design:** Develop initial UI designs and prototypes.
- **Feedback:** Collect feedback on prototypes from stakeholders and users.

Phase 2: Development

- **Implementation:** Develop the new UI based on approved designs.
- **Internal Testing:** Conduct initial rounds of functional, usability, and performance tests internally.

Phase 3: Beta Testing

- **Beta Release:** Release the new UI to a select group of users for beta testing.
- **Feedback Collection:** Gather feedback from beta users regarding functionality, usability, and performance.
- **Bug Fixes:** Address any issues identified during beta testing.

Phase 4: Final Testing and Optimization

- Conduct thorough testing to ensure all issues are resolved.
- Optimize the UI for performance and usability based on test results.

Phase 5: Launch

- Perform final checks and ensure all systems are ready for launch.
- Launch the new UI to all users.

Phase 6: Post-Launch

- Continuously monitor the performance and user feedback.
- Regularly release updates and improvements based on user feedback and performance data