LOGICLOOM FEEDBACK SYSTEM

REVOLUTIONIZING CUSTOMER VOICES

Objectives

1) Software Model
2) Functional Requirement
3) Non-Functional Requirement
4) Summary
5) Questions and Answers
6) Testing
7) Feedback form
8) Conclusion

1) Agile Software Development Model

Sprint Planning (Agile):

Divide the development into short sprints.

Prioritize features, focusing on core functionalities like user feedback forms, ratings, and data storage.

Development (Agile/Scrum):

Utilize Node.js for the backend development, creating a responsive and efficient server.

Implement MongoDB for data storage, taking advantage of its flexibility for handling diverse data types.

Design and develop the user interface for the feedback form using a front-end framework (e.g., React).

Regularly integrate code changes and conduct automated tests to ensure the reliability of the system.

User Feedback Loop (Agile):

Continuously gather feedback from potential users during development to make iterative improvements.

Admin Dashboard Development (Agile/Scrum):

Build the admin dashboard using a combination of Node.js for the backend and a suitable front-end framework.

Integrate Python for handling the interpretation of form data and connect to the ChatGPT API for enriched insights.

Data Visualization (Agile):

Develop the line graph to showcase the geographical distribution of user feedback.

Implement bar graphs to represent product ratings.

Design a pie chart for visualizing the percentage of users recommending the product.

User Acceptance Testing (Agile/Scrum):

Regularly involve stakeholders in testing to ensure that the evolving product meets their expectations.

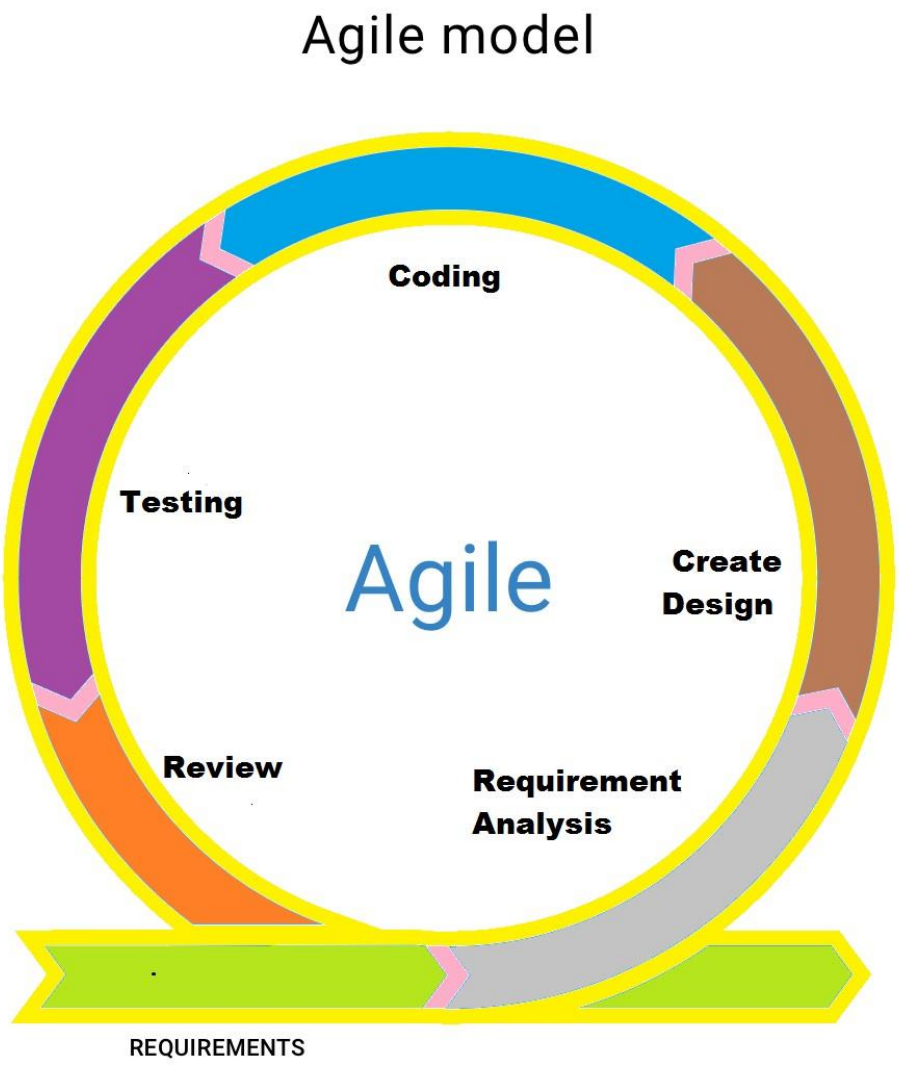
Deployment (Agile):

Deploy new features incrementally, ensuring that each release is stable and functional.

Feedback and Iteration (Agile):

Gather feedback from users and administrators after deployment to inform future iterations and improvements.

Agile Model (Scrum)



2) Functional Requirement

User Feedback Form:

The website should provide a user-friendly feedback form with fields for name, email address, country, feedback, product rating, and product recommendation.

Data Validation:

Implement validation checks to ensure that users provide necessary information in the required format. Validate email addresses to ensure correctness.

Rating System:

Include a rating system allowing users to rate the product on a scale of excellent, very good, good, fair, and poor.

Database Integration (MongoDB):

Establish a connection to MongoDB to store and retrieve user feedback data.

Admin Dashboard:

Create an admin dashboard to view and manage the submitted feedback.

Display a line graph indicating the distribution of feedback from different countries.

Graphs and Charts:

Implement bar graphs representing the distribution of product ratings (excellent, very good, good, fair, poor). Include a pie chart illustrating the percentage of users who recommended the product and those who did not.

3) Non-Functional Requirement

Scalability:

Design the system to handle potential increases in user traffic and data volume.

Performance:

Optimize the website and database queries for quick response times, ensuring a smooth user experience.

Reliability:

Implement error handling mechanisms to ensure the system remains stable and reliable.

User Experience (UX):

Design an intuitive and visually appealing user interface for the feedback form and admin dashboard.

Compatibility:

Ensure cross-browser compatibility and responsiveness for a seamless experience on various devices.

Data Privacy and Security:

Implement secure data transmission and storage practices to protect user information.

Backup and Recovery:

Set up regular backups of the MongoDB database to prevent data loss and ensure quick recovery in case of any issues.

Documentation:

Provide comprehensive documentation for developers, administrators, and end-users.

Integration Testing:

Conduct thorough testing to ensure proper integration between Node.js, MongoDB, Python, and the ChatGPT API.

Regulatory Compliance:

Ensure compliance with relevant data protection and privacy regulations in the regions where the website is accessible.

4) Summary

LogicLoom NPS Feedback Mechanism: Elevating User Engagement with Seamless Feedback Submission and Advanced Analytics!

LogicLoom, developed for DevRev's latest project, presents a cutting-edge NPS feedback mechanism that seamlessly integrates with Node.js and MongoDB to provide an unparalleled user experience. Users can submit their feedback, including their name, email address, country, and a comprehensive rating system, allowing them to rate the product on a scale from excellent to poor and indicate whether they would recommend it.

This innovative platform not only captures user input but also ensures data integrity and security by storing all information in a MongoDB database. The submitted forms are efficiently transferred to the admin in an interpreted form using Python, leveraging an integrated ChatGPT API for enhanced communication.

The admin dashboard is a powerhouse of analytics, featuring a dynamic line graph showcasing user participation from different countries. In addition, the dashboard displays user-specific details such as name, email, and a bar graph illustrating the distribution of product ratings (excellent, very good, good, fair, poor). Furthermore, a pie chart visually represents the percentage of users who recommended the product versus those who did not.

LogicLoom not only streamlines the feedback process but also empowers administrators with comprehensive insights, enabling them to make data-driven decisions for continuous product improvement and customer satisfaction.

5) Questions and Answers

Q: What is LogicLoom NPS Feedback Mechanism?

A: LogicLoom NPS Feedback Mechanism is a sophisticated website developed by DevRev that allows users to provide feedback on products through an intuitive dashboard. The system is built using Node.js and utilizes MongoDB to store all the feedback data.

Q: What information is collected in the feedback form?

A: The feedback form captures the user's name, email address, country, their feedback, their rating of the product (excellent, very good, good, fair, or poor), and whether they would recommend the product (yes or no).

Q: How is the data stored in LogicLoom?

A: All the data collected from the feedback forms is stored in a MongoDB database, ensuring a robust and scalable storage solution.

Q: How is the submitted feedback transferred to the admin?

A: The submitted feedback is transferred to the admin in an interpreted form using Python. This process involves integrating the ChatGPT API to enhance the interpretation of the feedback.

Q: What features does the admin dashboard offer?

A: The admin dashboard provides a comprehensive overview, including a line graph depicting the countries from which users have submitted feedback. It also displays user-specific information such as name, email, along with a bar graph illustrating the distribution of product ratings (excellent, very good, good, fair, poor). Additionally, a pie chart showcases the percentage of users who recommended the product and those who did not.

Q: How is the dashboard able to show country-wise data?

A: The system utilizes the geographic data collected from users to create a line graph that visually represents the distribution of feedback across different countries.

Q: Can the admin interact with the feedback data?

A: Yes, the admin has the capability to interact with the feedback data through the interpreted forms, making it easy to understand and respond to user feedback effectively.

Q: How does the system handle user ratings and recommendations?

A: The system processes and aggregates user ratings to generate a bar graph representing the distribution of ratings. It also uses the recommendation data to create a pie chart illustrating the percentage of users who recommend the product and those who do not.

Q: Is LogicLoom NPS Feedback Mechanism customizable?

A: Yes, the system is designed to be customizable, allowing for easy adaptation to specific project requirements or changes in the future.

6) Testing

Creating a robust testing strategy for the LogicLoom NPS (Net Promoter Score) feedback mechanism website, developed by DevRev using Node.js and MongoDB, involves comprehensive testing at various levels such as unit testing, integration testing, and system testing. Here's a breakdown of the testing approach for each level:

Unit Testing:

Test individual components, functions, and modules of the Node.js application.

Ensure that functions for handling user feedback, rating calculation, and data storage in MongoDB work correctly.

Integration Testing:

Verify the seamless integration between different components of the Node.js application.

Test the connection between the Node.js server and the MongoDB database to ensure proper data storage and retrieval.

Confirm that the integration with the ChatGPT API for interpreting submitted feedback forms is functioning correctly.

System Testing:

Evaluate the overall functionality of the NPS feedback mechanism website.

Test the complete flow of the feedback process, from users submitting forms to data interpretation by ChatGPT and storage in MongoDB.

Validate that the admin dashboard accurately displays the line graph representing user feedback distribution by country, along with user details and feedback statistics.

User Interface (UI) Testing:

Ensure a user-friendly experience by testing the dashboard's functionality, responsiveness, and overall design.

Validate that users can submit feedback forms easily and view their submitted data correctly.

Performance Testing:

Evaluate the website's performance under different load conditions to ensure it can handle multiple concurrent users.

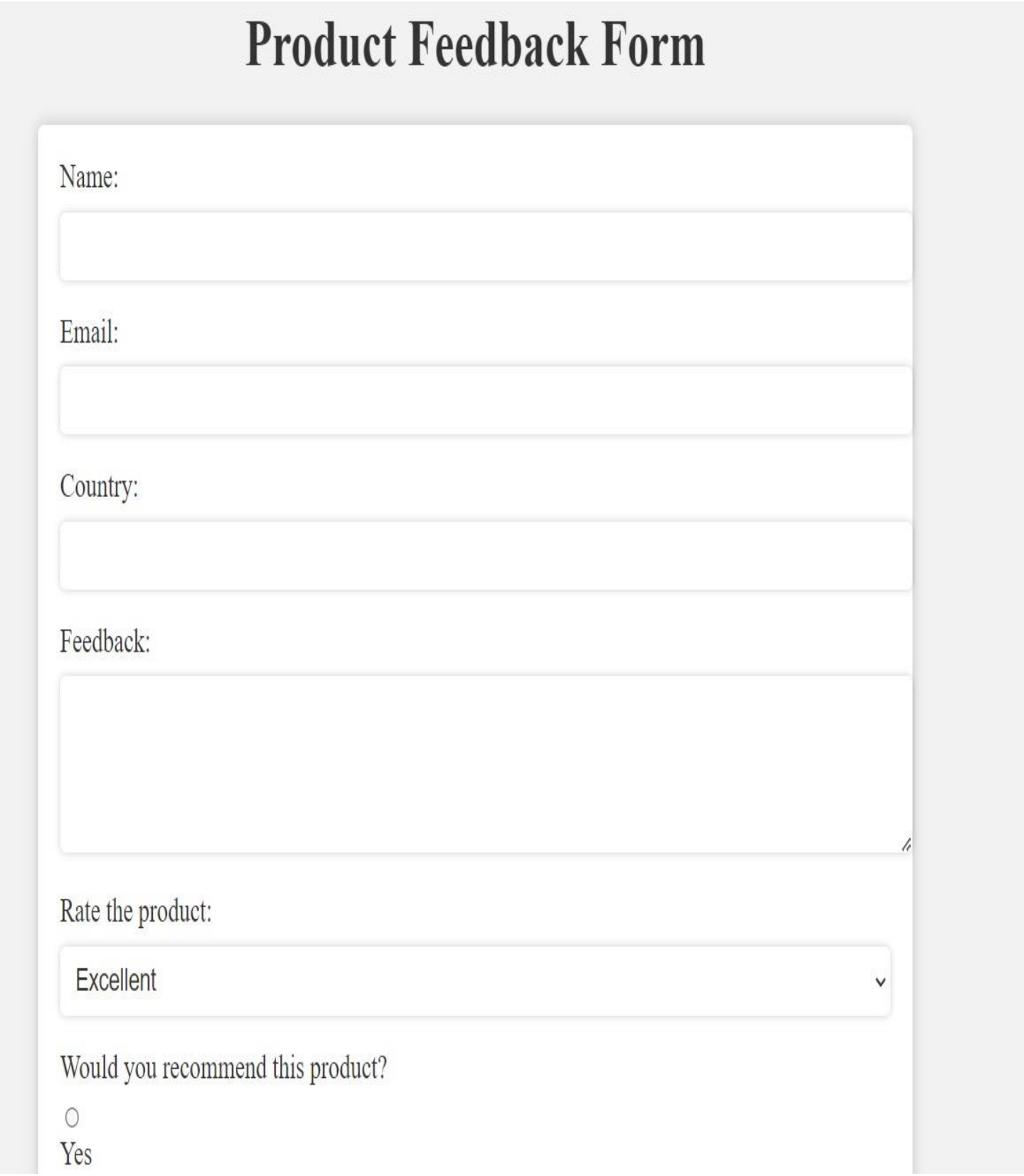
Check response times for form submissions, data processing, and dashboard updates.

Security Testing:

Perform security assessments to identify and fix vulnerabilities in the application.

Verify that user data is securely stored in the MongoDB database and that sensitive information is appropriately handled.

7) Feedback form



8) Conclusion

In conclusion, LogicLoom's NPS Feedback Mechanism, powered by an innovative blend of Node.js and MongoDB, stands as a pinnacle in usercentric product evaluation. This cutting-edge platform not only empowers users to provide detailed feedback but also ensures transparency and accountability with the inclusion of user information. The seamless integration of Python and ChatGPT API for interpreting and presenting data to the admin adds an extra layer of sophistication to the project.

The admin dashboard is a testament to the comprehensive insights this system offers, with its dynamic line graph showcasing user feedback distribution across different countries. The detailed breakdown of user information, coupled with insightful bar graphs illustrating product ratings, provides a nuanced understanding of product reception. The inclusion of a pie chart summarizing user recommendations adds a compelling visual element, making it easy for administrators to grasp the overall product sentiment.

In essence, the LogicLoom NPS Feedback Mechanism not only streamlines the feedback process but elevates it to a strategic tool for informed decision-making. This project exemplifies the fusion of cuttingedge technology and user-centric design, making it an indispensable asset for product development and enhancement.

Thank You!