**Project Documentation for Pre-Interview Project Submission**

**Web Feedback & Troubleshooting System Development**

Prepared By: Sai Arunabh Roy

Submission Date: 22nd April 2024

Application Link: [Feedback Management App](https://web-dev.software/)

1. **Introduction**
   1. *Project Objectives*

This project involved the creation of a dynamic feedback application designed to streamline user interaction and enhance backend processing using a variety of web development technologies. The primary objective was to build a robust platform where users can submit feedback, which is then processed and managed efficiently, providing actionable insights to administrators. This project demonstrates my capabilities in several areas crucial for the Web Developer role at the University of Utah Health like working with Gloss Design System, Website Troubleshooting, Bug Fixing and Code Optimization, Testing and Quality Assurance, Team Collaboration and Ticket Management.

* 1. *Key functionalities of the project include:* 
     + *Real-Time Interaction:* Leveraging web technologies to ensure immediate communication between users and administrators, crucial for timely feedback management.
     + *Problem-solving and Troubleshooting:* Efficiently diagnosing and resolving issues within the application, ensuring smooth functionality across various components.
     + *Web Development:* Utilizing ***Gloss Design System elements, HTML, CSS, JavaScript, PHP, and Node.js*** to create a responsive and user-friendly interface.
     + *Backend Processing:* Implementing server-side logic with ***Node.js and PHP*** to handle form submissions, database interactions, and email notifications.
     + *Database Management:* Integrating a ***MySQL*** database for robust data handling and retrieval, essential for managing large volumes of user feedback.
     + *Database Automation:* Utilizing a ***Python*** script to automatically create SQL dumps for easy database setup and management.
     + *Heroku Cloud Hosting:* Deploying the application on Heroku to demonstrate proficiency in ***cloud-based hosting environments***, ensuring scalable and accessible web application management.
  2. *Project Motivation:*

This project aligns perfectly with the responsibilities outlined for the Web Developer position at the University of Utah Health, particularly in troubleshooting, developing, and managing web applications using cutting-edge technologies. By implementing this project, I have demonstrated a strong foundation in both front-end and back-end development, database management, and real-time web communication—skills that are essential for success in this role.

1. **Project Background**

The initial concept for this feedback tool was conceived to address a critical need within web-based environments—efficient management and processing of user feedback. In many organizations, feedback systems are often fragmented or inefficient, leading to delays in response and potential dissatisfaction among users. This project aimed to create a streamlined, user-friendly platform that not only collects feedback effectively but also facilitates swift communication between users and administrators.

* 1. *Context and Relevance to Web Development*
     + ***User-Centric Design:*** At the heart of modern web development is the user experience, and this project serves as a testament to building applications that prioritize user interaction and satisfaction. The feedback tool is designed to be intuitive and accessible, ensuring that users can easily navigate and utilize the platform without any technical difficulties.
     + ***Real-Time Data Interaction:*** The use of real-time data processing ensures that feedback submitted by users is immediately available to administrators. This feature is crucial for web applications in dynamic environments where timely information exchange is key to effective decision-making and user engagement.
     + ***Comprehensive Feedback Management:*** By integrating features that allow for immediate alerting and response, the system aligns with current trends in web development that favor interactive and responsive designs. It supports a proactive approach to user management and service improvement.
  2. *Integration with Broader Systems*
     + ***Scalability and Flexibility:*** The application is built with scalability in mind, using Heroku for cloud hosting, which allows for easy scaling as user demand grows. This is particularly important for institutions like the University of Utah Health, where user feedback can be voluminous and variable.
     + ***Security and Reliability:*** Utilizing established technologies such as PHP for server-side scripting and MySQL for database management ensures that the application is not only robust and capable of handling large volumes of data but also secure against common web vulnerabilities.

This project serves as a bridge between technical expertise and practical application, applying the principles of modern web development while addressing a need within any service-oriented institution. It sets a foundation for continuous improvement and adaptation, traits that are invaluable to the evolving landscape of web technologies.

1. **Technologies Used**

In the development of the feedback tool, a variety of modern technologies, languages, and frameworks were utilized to ensure robust functionality and seamless user interaction. Here's a breakdown of each component.

* 1. *Frontend Technologies*
     + ***HTML****:* Used to structure the content on the web pages, providing the skeleton for forms, buttons, and text fields.
     + ***CSS & GLOSS Design System****:* The styling was significantly enhanced by adhering to the GLOSS Design System, which is a cornerstone of web development at the University of Utah Health. This system provides a consistent and accessible user interface, aligning the application's aesthetic with the University’s branding guidelines. CSS was employed to style the presentation of the HTML content, ensuring the interface is visually appealing and user-friendly. The use of responsive design principles ensures that the application is accessible on a variety of devices, from desktops to mobile phones.
     + ***JavaScript****:* Facilitates dynamic interactions on the client side, enhancing the user experience by making the application interactive and responsive in real time.
  2. *Backend Technologies*
     + ***PHP:*** A server-side scripting language used to create dynamic web pages and handle the business logic of the application. PHP was particularly instrumental in processing user feedback submissions and communicating with the MySQL database.
     + ***Node.js:*** Utilized for handling asynchronous events, particularly in sending email notifications through the *Nodemailer* package. This allowed for real-time communication between the application and users, enhancing the feedback loop.
  3. *Database Management*
     + ***MySQL:*** Hosted on Heroku, this relational database management system stores all user feedback and associated metadata securely. It provides robust data retrieval capabilities that are essential for generating reports and analyzing user feedback efficiently.
     + ***Python:*** Leveraged Python libraries like *random* and *json* for scripting to automate and generate the SQL dump for mass data imports.
  4. *Version Control and Deployment*
     + ***Git:*** Used for version control, allowing for incremental updates to the application while maintaining a comprehensive history of all changes. This is crucial for collaborative environments and ensures that the development process is streamlined and error-free.
     + ***Heroku:*** A cloud platform service that supports the hosting and scaling of web applications. Heroku was chosen for its ease of use, scalability, and integration with existing development workflows, such as GitHub.
  5. *Testing and Quality Assurance*
     + ***Postman:*** Utilized for testing the APIs developed for the backend to ensure they handle requests and responses correctly.
     + ***Mocha:*** Employed for server-side code testing to validate the functionality and reliability of the backend operations, particularly in scenarios involving data processing and email notifications.

The combination of these technologies provided a comprehensive environment for developing a highly functional and responsive feedback tool. Each technology choice was guided by the specific needs of the project—from user interface design to data management and server-side processing—ensuring that the tool not only meets the current requirements but is also scalable for future enhancements. The integration of the GLOSS Design System was pivotal in aligning the application's user interface with the university’s visual identity standards, emphasizing its role in maintaining a coherent user experience across university web platforms.

1. **Project Features and Justification**

The feedback / issue reporting tool was designed with several key features that align closely with the responsibilities and qualifications of a Web Developer at the University of Utah Health. Each feature was chosen not only for its functional benefits but also to demonstrate competence in various aspects of web development that are critical for the role.

* 1. *Feedback Form Integration*
     + *Purpose:* The primary interface for users to submit their feedback or report issues.
     + *Technology:* Implemented using HTML forms, CSS for styling, and JavaScript for front-end validation.
     + *Justification:* Demonstrates proficiency in frontend development, ensuring user inputs are correctly captured and validated before submission. This is essential for reducing errors and improving data quality, a critical skill for any developer focused on user experience and interface design.

A screenshot of a survey

Description automatically generated

* 1. *Email Notifications*
     + *Purpose:* To notify users and administrators upon the successful submission of feedback, and ticket completion.
     + *Technology:* Utilizes Node.js and the Nodemailer library to send automated emails.
     + *Justification:* Showcases backend skills and understanding of server-side programming. Automated emails are crucial for confirming user actions and keeping records within the team, which improves the communication flow and user satisfaction.

A screenshot of a feedback form

Description automatically generated

*A close-up of a contact us

Description automatically generated*

* 1. *Database Management*
     + *Purpose:* To securely store and manage user feedback data.
     + *Technology:* MySQL database hosted on Heroku.
     + *Justification:* Handling data with a robust management system is vital for web applications. It allows for scalable solutions and provides the developer with experience in managing complex datasets—skills that are paramount for the job’s requirements related to database and server management.

A screenshot of a computer

Description automatically generated

* 1. *Error Handling and Debugging*
     + *Purpose:* To ensure that the application runs smoothly and any issues are promptly resolved.
     + *Technology:* Implemented through comprehensive error-checking in both PHP and JavaScript.
     + *Justification:* Reflects the ability to troubleshoot and resolve issues efficiently, an essential duty in the job description. This feature not only helps in maintaining the integrity of the application but also improves user trust and application reliability.

*A screen shot of a computer program

Description automatically generated*

* 1. *Responsive Design & Navigation*
     + *Purpose:* To make the application accessible on various devices with different screen sizes.
     + *Technology:* CSS Media Queries and GLOSS Design System guidelines.
     + *Justification:* Ensures that the application can be used effectively across all platforms, which is crucial for the broad audience of the University of Utah Health. This aligns with the job’s requirement for a developer capable of creating flexible, user-centric designs.

*A close up of a logo

Description automatically generated*

*A close-up of a cover

Description automatically generated* *A screenshot of a search box

Description automatically generated*

* 1. *Dynamic Searching, Sorting, and Filtering*
     + Purpose: To enhance the user experience by providing flexible and efficient ways to view and manage feedback entries.
     + Technology: Implemented using JavaScript for dynamic client-side interactions and PHP for server-side data processing.
     + Justification: This feature allows administrators to easily locate specific feedback entries based on various criteria such as issue type, status, or user information, which is crucial for effective data management. Sorting capabilities enable quick organization of data in a meaningful order, while filtering options help focus on entries that meet certain conditions, reducing the time spent searching for information.

A screenshot of a search engine

Description automatically generated

* + - Dynamic Searching offers real-time results as the user types, minimizing the need for full-page refreshes and enhancing user satisfaction by providing immediate feedback.
    - Sorting functionality allows users to order feedback based on different attributes such as date, name, or status. This is particularly useful in managing large volumes of data, making the system more user-friendly and efficient.
    - Filtering options empower users to display only those records that meet specific criteria, thus simplifying the management tasks and allowing for better focus on issues or feedback types.

These features collectively enhance the functionality of the feedback tool, making it a robust application for managing user interactions and data. By integrating these features, the project not only serves its primary function but also acts as a testament to the developer’s skills in aligning with the specific needs of the University of Utah Health’s Web and Interactive Marketing department.

1. **Challenges Overcome**

Throughout the development of the feedback management system, several challenges were encountered that required thoughtful solutions and adaptations. These challenges not only tested my technical skills but also provided valuable lessons in web application development.

* 1. *Database Integration on Heroku*
     + *Challenge:* Initially integrating the MySQL database with the Heroku-hosted application proved challenging due to Heroku's transient filesystem and connectivity.
     + *Solution:* I utilized Heroku's add-on services to set up a managed MySQL database, ensuring persistent storage and robust database management. Configuring the environment variables and modifying connection strings in the PHP scripts to adapt to Heroku's dynamic environment were key steps that resolved the issues.
  2. *Email Notification Reliability*
     + *Challenge:* Ensuring reliable delivery of email notifications via Nodemailer with Zoho SMTP was problematic due to authentication issues and SMTP server configurations.
     + *Solution:* By switching to environment variables for sensitive credentials and debugging the SMTP settings, I established a secure and reliable email delivery system. This involved detailed error logging and iterative testing to refine the setup.
  3. *Real-time Data Interaction*
     + *Challenge:* Creating a real-time dynamic interaction between the admin and the users for updates on feedback status required a seamless backend and frontend integration.
     + *Solution:* I implemented AJAX calls and server-side APIs that allowed the frontend to fetch and display updated data without reloading the page. This asynchronous communication was crucial for maintaining a smooth user experience.
  4. *Use of GLOSS Design System*
     + *Challenge:* Adhering to the GLOSS Design System specific to the University of Utah Health posed constraints on design flexibility.
     + *Solution:* By thoroughly studying the design guidelines and integrating them with the application’s UI, I ensured that the visual elements conformed to the institutional branding while maintaining functional integrity. This was achieved by customizing CSS stylesheets to align with GLOSS specifications without compromising on the application’s usability.
  5. *Dynamic Content Management*
     + *Challenge:* Implementing dynamic searching, sorting, and filtering that responded to user inputs in real-time was initially inefficient.
     + *Solution:* Optimizing JavaScript functions for searching and sorting, and leveraging PHP for efficient data retrieval operations, dramatically improved performance. Utilizing AJAX for partial page updates reduced load times and enhanced the user experience.

These challenges were instrumental in deepening my understanding of full-stack development, from client-side interactions to server-side processing and database management. Each resolved issue enhanced the application’s functionality and reliability, directly contributing to a robust and user-friendly interface.

1. **Results and Impact**

The deployment of the feedback management system significantly enhanced the way feedback is collected and managed, demonstrating the practical utility of the application in a real-world context. Here’s an overview of the impact and results of the project.

* 1. *Enhanced User Experience*
     + ***Immediate Feedback Confirmation:*** Users receive instant email confirmations after submitting their feedback, ensuring them that their input has been received and is being processed. This feature, powered by Nodemailer and SMTP integration, has improved user satisfaction and trust in the feedback mechanism.
     + ***Real-Time Updates:*** The dynamic data interaction capabilities allow users to see real-time status updates on their feedback directly on the platform. This keeps users informed and engaged with the process, reducing the need for follow-up inquiries.
  2. *Administrative Efficiency*
     + ***Streamlined Feedback Management:*** Admins can now easily access, sort, filter, and manage feedback entries through a well-organized dashboard. This includes capabilities like dynamic searching and filtering which allow for efficient handling of large volumes of data.
     + ***Quick Issue Resolution:*** With structured data and clear categorization (using the GLOSS Design System), admins can quickly identify and address issues. The ability to sort feedback based on severity or date ensures that urgent matters are prioritized.
  3. *Innovative Features*
     + ***JSON Comment Conversion:*** The system cleverly converts JSON-formatted comments into readable conversation texts that are included in the closing ticket emails. This feature not only streamlines communication but also provides clear, conversational context about the feedback progression, which is crucial for both administrative review and user satisfaction.
  4. *Technical Robustness*
     + ***Database Reliability:*** The integration with a MySQL database hosted on Heroku provides a stable and scalable backend. This ensures that feedback data is securely stored and readily accessible, even as the volume of user data grows.
     + ***Cross-Platform Compatibility:*** Extensive testing across different devices and browsers guarantees that the application performs reliably for all users, regardless of their access point. This universal accessibility is crucial for inclusivity and user reach.

1. **Future Enhancements**

Looking forward, the feedback management system can be expanded in many ways to enhance functionality and user experience.

* Mobile App Integration: Developing a mobile application to provide another streamlined channel for feedback submission and tracking.
* AI-Driven Insights: Implementing machine learning algorithms to analyze feedback trends and predict areas of concern, allowing preemptive action.
* Enhanced Security Features: Increasing data protection measures to ensure user information is meticulously safeguarded against emerging cyber threats.

1. **Conclusion**

Completing this project has been a comprehensive and enriching experience that not only reinforced my web development skills but also aligned closely with the responsibilities and expectations of a Web Developer at the University of Utah Health. This initiative allowed me to effectively demonstrate my capability to troubleshoot, design, and enhance web applications, underscoring my readiness to contribute positively to the Web and Interactive Marketing team.

* 1. *Skills and Project Outcomes*
     + ***Technical Proficiency:*** The project utilized a wide range of technologies—HTML, CSS, JavaScript, PHP, Node.js, MySQL, and various deployment and version control tools. The successful integration of these technologies to build a fully functional feedback management system justifies my technical stack and my ability to apply these skills in a cohesive and practical manner.
     + ***Problem-Solving Efficiency:*** Throughout the project, I encountered and overcame multiple challenges, such as database connectivity issues, email delivery reliability, and dynamic content management. Tackling these obstacles has improved my troubleshooting skills and my ability to implement effective solutions under pressure.
     + ***Design and Usability Focus:*** By adhering to the GLOSS Design System, I ensured that the application not only met aesthetic standards but also provided a user-friendly experience, crucial for both the administrative backend and the user-facing frontend. This adherence highlights my attention to detail and commitment to delivering quality user experiences.
  2. *Alignment with Career Goals*

***This project is a step towards my goal of becoming a Senior Developer****,* as it involved elements of project planning, execution, and management—skills necessary for advanced roles. The experience has prepared me for the multifaceted challenges of web development and system optimization in a large-scale health care setting.

* 1. *Commitment to Continuous Improvement*

I am committed to ongoing professional development to keep abreast of the latest technologies and methodologies in web development. This project has elevated my confidence in my abilities and my potential for future growth within the University of Utah Health's innovative environment.

1. **Appendix**

This section of the document serves as a repository for supplementary materials. These include references to third-party libraries , APIs and documentation used throughout the project.

* 1. *Third-Party Libraries and APIs*
     + ***Nodemailer:*** Used for handling outbound emails seamlessly.
     + ***Express.js:*** A web application framework for Node.js, used to simplify routing and middleware functionality.
     + ***MySQL:*** Database management for storing and retrieving user feedback data.
     + ***Body-Parser:*** Middleware for parsing incoming request bodies in a middleware before your handlers, available under the req.body property.
     + ***CORS (Cross-Origin Resource Sharing***): A node.js package used to provide a Connect/Express middleware that can be used to enable CORS with various options.
  2. *Documentation References*
     + ***GLOSS Design System Documentation:*** Provides guidelines and standards for web development within the University of Utah Health system.
     + ***Heroku Official Documentation:*** For deployment procedures and best practices.
     + ***Node.js Documentation:*** Essential for understanding backend development nuances and best practices, particularly in relation to asynchronous programming and server setup.
     + ***MDN Web Docs:*** A valuable resource for frontend development, offering detailed documentation on HTML, CSS, and JavaScript.
     + ***PHP Official Documentation:*** Crucial for backend developers using PHP, providing extensive information on syntax, functions, and best practices.
     + ***MySQL Documentation:*** In-depth resource for database administrators and developers, detailing SQL syntax, query optimization, and configuration.
  3. *Additional Resources*
     + ***Postman Documentation:*** Guides on using Postman for API testing, which helps in ensuring that APIs are working as expected before they are deployed or integrated into front-end applications.
     + ***Git Documentation:*** For version control, essential for managing changes to the project codebase, collaborating with other developers, and integrating continuous deployment pipelines.
     + ***W3C Validator:*** A tool for validating HTML and CSS markup, ensuring compliance with web standards and accessibility guidelines.
     + ***Stack Overflow and GitHub Communities:*** For troubleshooting and community support, these platforms offer discussions, solutions, and code snippets shared by developers around the world.

***Disclaimer***

*The use of the University of Utah Health's brand guidelines and logo in this project is strictly for the purpose of this pre-interview project submission and is not intended for personal or commercial use.*

*End of Documentation*