Chapter 4

System Implementation

**4.1 System Implementation**

System implementation and testing are integral stages in developing any software, including systems like the one you are working on. These stages typically involve several key steps such as design, development, testing, and deployment. In the design phase, the system's requirements and architecture are defined. The development phase involves coding and integrating various components of the system. Testing ensures the system's functionality, security, and performance, while the deployment phase marks the system's readiness for real-world use. Each of these steps is crucial for creating a robust and reliable system.

Design Phase: In the web-based liver cirrhosis detection system, the design of the user interface and user experience is specifically customized to fit a user’s need for detection and classification of the cirrhosis disease. In the design phase of the web-based liver cirrhosis detection system, the primary focus is on crafting a user interface and experience tailored to the specific requirements of users seeking to detect and classify cirrhosis disease. This involves creating an intuitive UI with clear navigation and visually appealing elements, coupled with a seamless UX that guides users through the detection process efficiently. Accessibility and inclusivity are paramount considerations, ensuring the system is usable by all, while robust security measures safeguard user data and compliance with relevant regulations. The design underwent rigorous testing and iteration to refine its effectiveness, culminating in a user-friendly and informative tool for informed decision-making regarding cirrhosis detection.

Development Phase: The development phase of the system utilized Streamlit, chosen for its ability to deliver a consistent user experience across various devices. Streamlit's cross-platform capabilities were instrumental in ensuring that users of all types, received the same functionality and visual experience, regardless of the device used. For backend operations, Python and Pytorch were integrated, providing a robust and secure platform for machine learning inference.

Testing Phase: In the web-based liver cirrhosis detection system's testing phase, a comprehensive strategy was implemented. Initially, unit testing was conducted to ensure each system component functioned correctly. Following this, integration testing was crucial to verify seamless interaction between the Streamlit front-end and Python backend, essential for overall system coherence. The final stage involved user interface testing, focusing on ensuring the system's ease of use, responsiveness, and intuitiveness across all user roles, from donors to administrators. This multi-level testing approach was pivotal in guaranteeing a robust, user-friendly, and fully functional system.

Deployment Phase: The deployment phase of the web-based liver cirrhosis detection system was methodically segmented into two key stages: staging and production deployment. The staging deployment served as the preliminary phase, where the system was deployed in a controlled environment. This stage was crucial for conducting final tests under conditions that closely simulated the actual operating environment, allowing for the identification and rectification of any residual issues. Following successful staging deployment, the system proceeded to production deployment. This stage marked the system's availability for public access, signifying its readiness for real-world application. During production deployment, continuous monitoring was employed to optimize performance and ensure smooth operation, with a focus on maintaining system reliability and efficiency in a live environment.

**4.2 Broad Description of The Application**

The web-based liver cirrhosis detection system is a powerful tool designed to assist in the diagnosis and classification of cirrhosis disease. Leveraging the capabilities of Python, PyTorch, and Streamlit, this system provides a user-friendly interface for both medical professionals and patients.

At its core, the system utilizes a PyTorch model trained on a dataset of liver images to classify images as either indicative of cirrhosis or not. This model is loaded and used in real-time to provide immediate feedback on uploaded images, making it a valuable aid in the diagnostic process.

The user interface is built using Streamlit, a popular framework for creating web applications with Python. Streamlit allows for easy integration of the PyTorch model, as well as the development of interactive elements such as file upload buttons and result displays. This ensures a seamless user experience and enables quick and intuitive use of the system.

In addition to its diagnostic capabilities, the system also provides prognostic services. This helps users better understand the disease and its implications, empowering them to make informed decisions about their health.

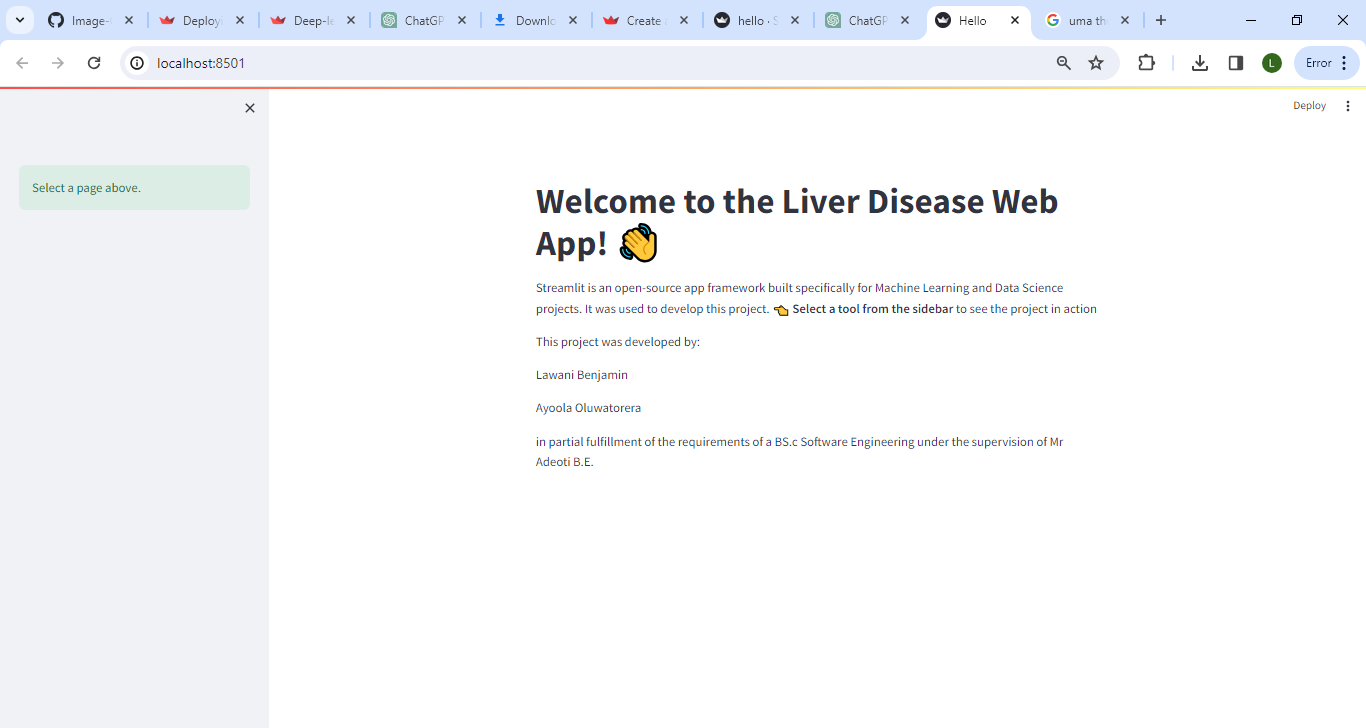
Overall, the web-based liver cirrhosis detection system represents a significant advancement in the field of medical imaging, offering a reliable and accessible tool for the detection and classification of cirrhosis disease.

**4.3 Implementation of The App**

The successful implementation of the web-based liver cirrhosis detection system is characterized by a user-centric approach. This section provides a comprehensive walkthrough of the application's functionalities, ensuring clarity and ease of use for a prospective user.

**4.3.1 Welcome Screen**

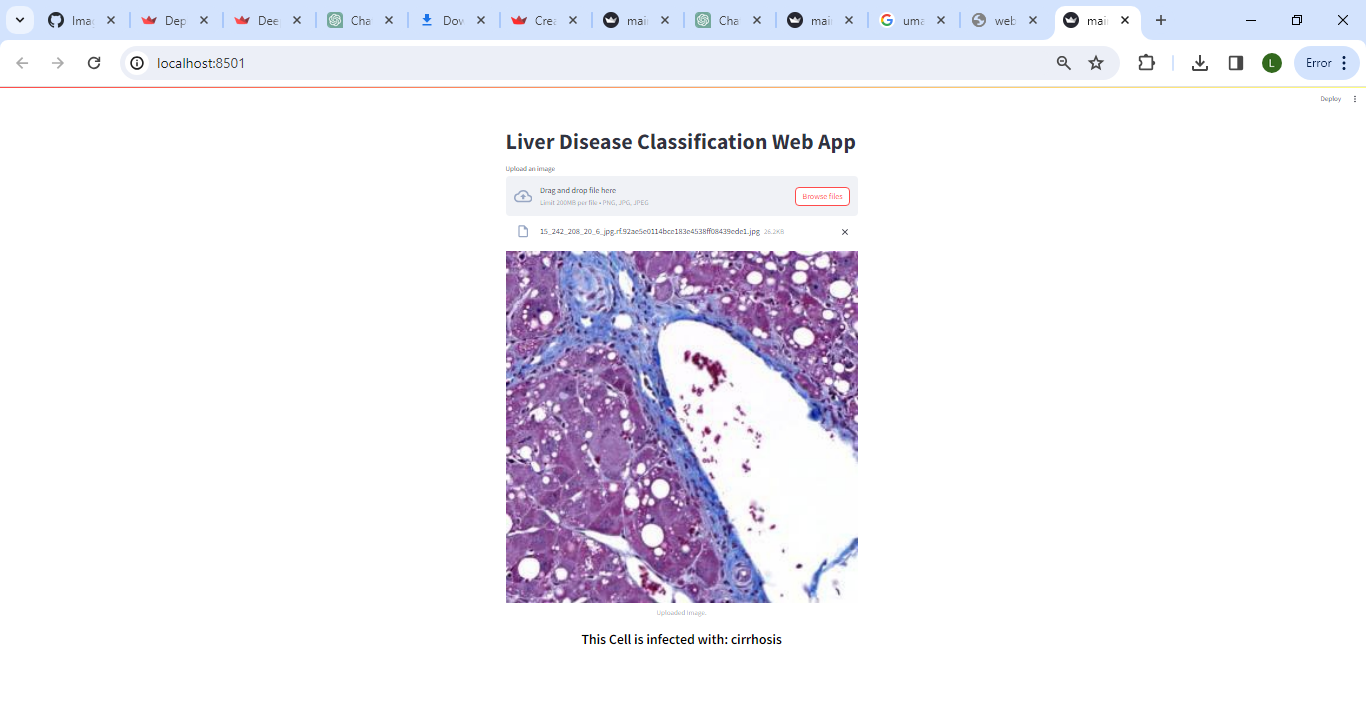
The user visits the website and is met with a user-friendly welcome screen



**Figure 4.1 Welcome Screen**

**4.3.2 Disease Classification**

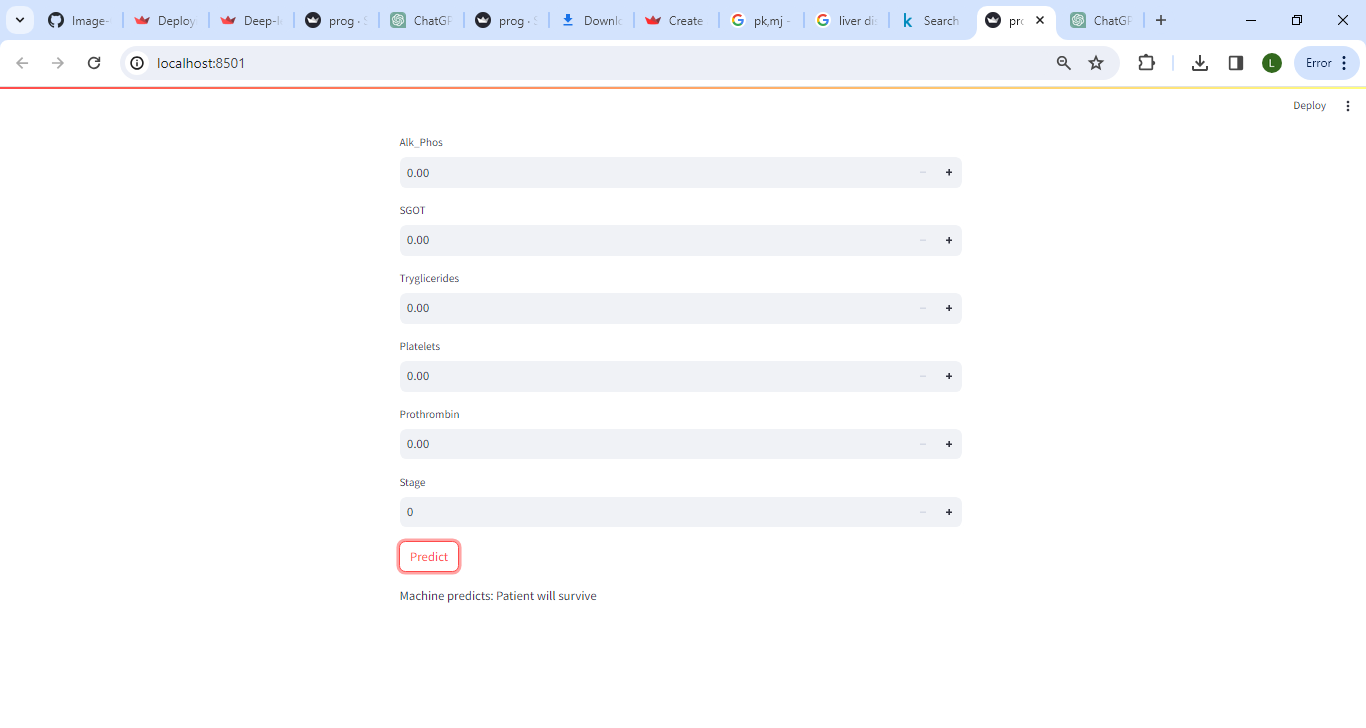
An image of a cell in question is uploaded into the upload area, the system confirms it is in the right file format, takes it in, processes it and infers if the cell is diseased with cirrhosis



**Figure 4.2 Disease Prediction Page**

**4.3.3 Prognosis Prediction**

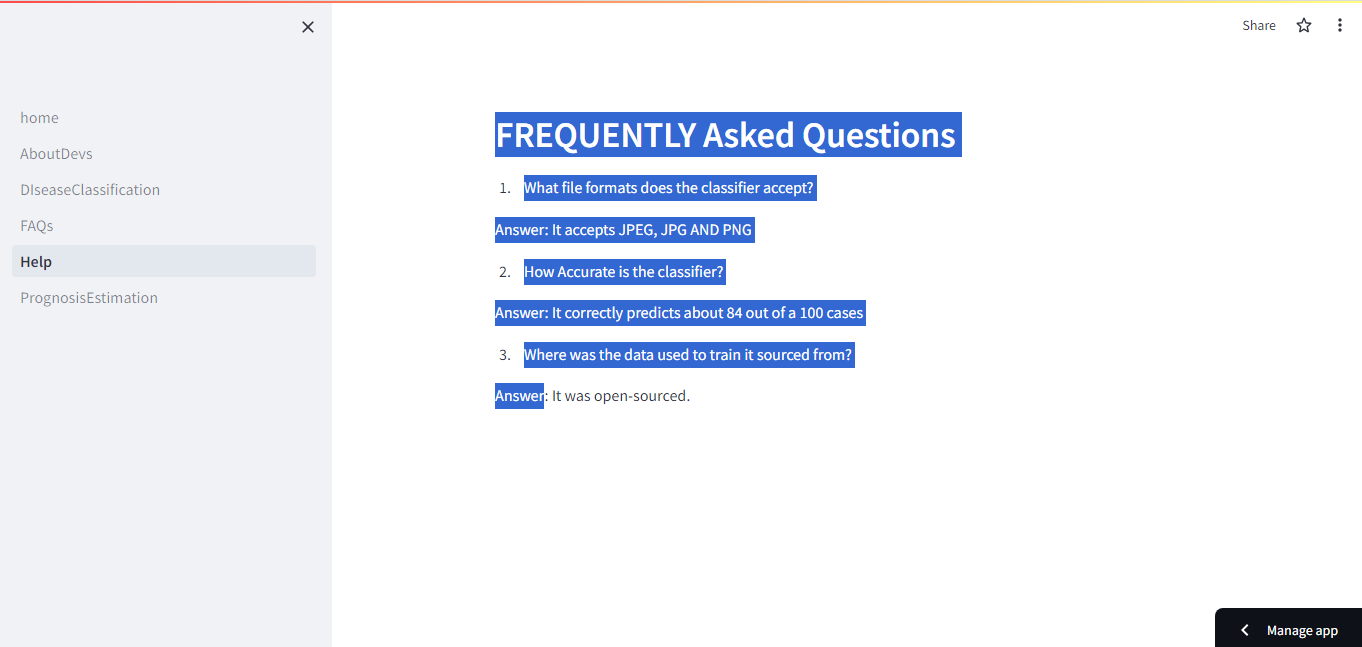
The user’s vitals are entered into the system, and the system predicts if the patient survives the disease or does not survive



**Figure 4.3 Prognosis Prediction Page**

**4.3.4 Frequently Asked Questions Page**

This page shows answers to questions often asked about the web app and its internal workings.



**Figure 4.4 Frequently Asked Questions Page**

**4.4 Results**

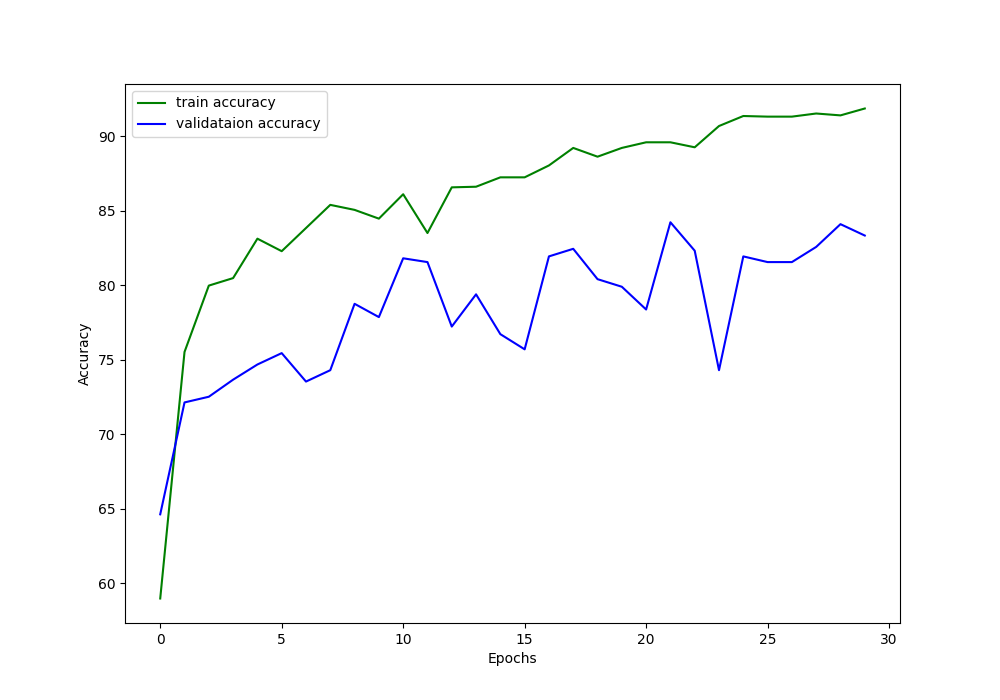
The web-based liver cirrhosis detection system has a profound impact on diagnostics, particularly in the field of liver disease. By leveraging advanced technologies such as Python, PyTorch, and Streamlit, the system offers several key benefits that enhance the diagnostic process.

Firstly, the system provides a fast and efficient means of diagnosing cirrhosis disease. With the ability to analyze images in real-time, medical professionals can quickly assess whether a patient's liver is indicative of cirrhosis, allowing for prompt treatment and management.

Secondly, the system improves the accuracy of diagnoses. The PyTorch model used in the system is trained on a large dataset of liver images, enabling it to recognize patterns and features indicative of cirrhosis with accuracy of around 84 percent. This reduces the likelihood of misdiagnosis and ensures that patients receive the appropriate care.

Additionally, the system enhances the accessibility of diagnostic tools. By providing a user-friendly interface, the system can be easily used by medical professionals with varying levels of expertise. This helps to democratize access to advanced diagnostic technologies, particularly in regions where access to healthcare resources may be limited.

Overall, the web-based liver cirrhosis detection system has a transformative impact on diagnostics, offering a fast, accurate, and accessible tool for the diagnosis and classification of cirrhosis disease.



**Figure 4.5 The Classifier’s Training and Validation Accuracy**

**4.5 Testing**

Testing is important to ensure the reliability, functionality, usability, and security of the discussion forum. The following testing methods were employed:

i. Unit testing: This involves testing each component individually to verify its accuracy and functionality. The modals were tested individually to ensure that the upload actions, predict actions and download actions were executed correctly.

ii. Integration testing: This involved testing the interactions between different components to ensure that they are compatible. This was well used in testing the various functionalities and how they work together. It was also used while testing the pretrained models to ensure smooth bi-directional communication between the client-side and server-side.

iii. System testing: This involved testing the overall system to verify its functionalities and performance. This was used to test the navigation, start-up time, and system response time.

**4.6 Security**

The decision to not store data in the web-based liver cirrhosis detection system is a critical security measure that helps protect patient privacy and confidentiality. By not retaining any uploaded images or diagnostic results, the system ensures that sensitive medical information is not stored or at risk of being accessed by unauthorized parties.

This approach to security is in line with best practices for handling medical data, as it minimizes the potential impact of data breaches or security vulnerabilities. It also helps to comply with regulations and standards which require strict measures to safeguard patient information.

Furthermore, not storing data simplifies the system's architecture and reduces the need for complex data management and security measures. This can lead to a more streamlined and efficient system overall, benefiting both users and developers.