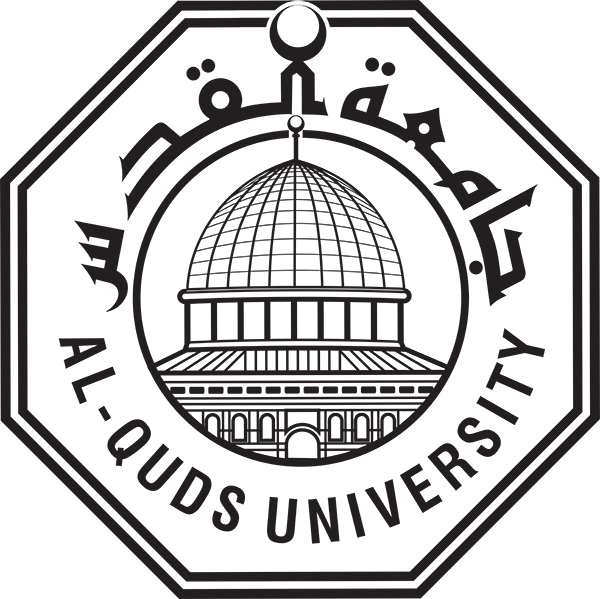
**Al-Quds University**

**Faculty of Science and Technology**

**Computer Science**



**AI-Integrated MRI Brain Tumor Detection Application**

**Prepared by:**

**Bishara Jack AlBabish**

**Saliba Elias Rishmawi**

**Supervised by**

**Prof. Awni Abu-Tair**

**Institution and Department:**

**Al-Quds Univeristy, Computer Science**

Submission Date:

20/202/2002

|  |  |
| --- | --- |
| List of Abbreviations: | |
| Abbreviations: | Definition: |
| AI  CNNs  CSS  DICOM  GUI  HTML  JavaScript  MRI  PDF  PyQt  PyTorch  QSS  Python | Artificial Intelligence.  Convolutional Neural Networks.  Cascading Style Sheets.  Digital Imaging and Communications in Medicine.  Graphical user interface interaction.  Hypertext Markup Language.  Web Development scripting language.  Magnetic Resonance Imaging.  Portable Document Format File.  Python bindings for the Qt libraries.  Open-source machine learning library.  Qt Style Sheets.  Programming language for automation. |

Contents

[Table of Figures 5](#_Toc186753756)

[Abstract 5](#_Toc186753757)

[Chapter One: Introduction 6](#_Toc186753758)

[1.1 Purpose 6](#_Toc186753759)

[1.2 Scope 7](#_Toc186753760)

[Chapter Two: Implementation 7](#_Toc186753761)

[2.1 Machine Learning Model (PyTorch) 7](#_Toc186753762)

[2.2 Project Documentation Website (HTML, CSS, JavaScript, Python) 7](#_Toc186753763)

[2.3 Workflow 8](#_Toc186753764)

[2.4 User Interface (PyQt) 8](#_Toc186753765)

[2.5 Team Contribution 9](#_Toc186753766)

[Chapter Three: Overall Description 10](#_Toc186753767)

[3.1 Product Perspective 10](#_Toc186753768)

[3.2 Product Functions 10](#_Toc186753769)

[3.3 User Classes and Characteristics 11](#_Toc186753770)

[Chapter Four: Specific Requirements 11](#_Toc186753771)

[4.1 Functional Requirements 11](#_Toc186753772)

[4.2 Non-Functional Requirements 12](#_Toc186753773)

[4.3 Interface Requirements 14](#_Toc186753774)

[Chapter Five: Diagrams 16](#_Toc186753775)

[5.1 Sequence Diagram 16](#_Toc186753776)

[5.2 Data Flow Diagram 17](#_Toc186753777)

[5.3 User Case Diagram 18](#_Toc186753778)

[5.4 Entity Relationship Diagram 18](#_Toc186753779)

[5.5 Class Diagram 18](#_Toc186753780)

[Chapter Six: Breakdown 18](#_Toc186753781)

[Chapter Seven: Prototype 19](#_Toc186753782)

[7.1 Header 19](#_Toc186753783)

[7.2 Introduction & Contact Us 20](#_Toc186753784)

[7.3 About Us 21](#_Toc186753785)

[Chapter Eight: Results and Discussion 29](#_Toc186753786)

[8.1 Survey Introduction 29](#_Toc186753787)

[8.2 Survey Questions 29](#_Toc186753788)

[8.2.1 General Information 29](#_Toc186753789)

[8.2.2 Experience with AI Applications 29](#_Toc186753790)

[8.2.3 User Interface and Experience: 29](#_Toc186753791)

[8.2.4 Data Privacy and Security: 30](#_Toc186753792)

[8.2.5 Feedback and Suggestions: 30](#_Toc186753793)

[8.3 Survey Analysis 30](#_Toc186753794)

[8.3.1 General Information: 30](#_Toc186753795)

[8.3.2 Experience with AI Applications: 30](#_Toc186753796)

[8.3.3 User Interface and Experience: 30](#_Toc186753797)

[8.3.4 Data Privacy and Security: 30](#_Toc186753798)

[8.3.5 Feedback and Suggestions: 30](#_Toc186753799)

[Chapter Nine: Future Work and Enhancements 30](#_Toc186753800)

[9.1 Additional Considerations 31](#_Toc186753801)

[Chapter Tenth: References 31](#_Toc186753802)

[Chapter Eleventh: Conclusion 31](#_Toc186753803)

Table of Figures**:**

[**Figure 4.1: HTML,CSS,JS Logos** 14](#_Toc186196941)

[**Figure 5.3: Sequence Diagram** 16](#_Toc186196942)

[**Figure 5.4: Data Flow Diagram** 17](#_Toc186196943)

[**Figure 5.5: Use Case Diagram** 18](#_Toc186196944)

[**Figure 7.1: Header of the Website** 19](#_Toc186753750)

[**Figure 7.2: Introduction of the Website** 20](#_Toc186753751)

[**Figure 7:3 Contact Us of the Website** 21](#_Toc186753752)

[**Figure 7.4: Bishara & Saliba Overview and Information** 24](#_Toc186753753)

[**Figure 7.5: How to Demo Demonstration** 25](#_Toc186753754)

[**Figure 7.6: Choosing a Picture from the folder** 28](#_Toc186753755)

# **Abstract**

This project focuses on developing an innovative AI-integrated application designed to analyze MRI scans of the brain and accurately detect the presence of tumors. Utilizing the PyTorch framework for its machine learning model, the project employs CNNs to achieve high accuracy in image analysis tasks. This approach ensures high performance and precision, addressing a critical need in medical diagnostics.

The application provides a user-friendly interface built with PyQt, designed exclusively for users to seamlessly upload MRI scans and receive real-time diagnostic results. Beyond identifying potential tumors, the application offers detailed information on tumor type, treatment options, and severity probabilities and others. Specifically targeting tumors such as **Glioma, No Tumor.**

The project utilizes the publicly available BraTS dataset from Kaggle for model training and validation. This dataset, known for its comprehensive and high-quality annotated MRI scans, ensures that the AI model is well calibrated and reliable for detecting various brain tumors.

The project also includes a comprehensive website developed using HTML, CSS, JavaScript, Python. The site documents the project in detail offering information about the AI model, application features, and a demonstration of the application’s capabilities.

Emphasizing privacy and user experience, the application is designed as a standalone tool with no web integration. Future enhancements include continuous model improvement, expansion of features. This project aims to significantly aid medical professionals and patients by providing an efficient, accurate, and user-friendly tool for brain tumor detection.

# **Chapter One: Introduction**

The early detection of brain tumors is a critical factor in improving patient outcomes and survival rates. MRI scans are a primary diagnostic tool used by medical professionals to identify and evaluate brain tumors. However, the process of analyzing these scans can be time-consuming and subject to human error. To address this challenge, our project aims to develop an AI-integrated application that leverages advanced machine learning techniques to assist in the accurate and efficient detection of brain tumors from MRI scans.

This project combines cutting-edge technologies in machine learning and user interface design to create a comprehensive diagnostic tool. By utilizing CNNs implemented in PyTorch, our AI model is capable of analyzing MRI images with high accuracy. The application is designed to be user-friendly, with a graphical interface built using PyQt that allows users to upload MRI scans and receive real-time diagnostic results. The results not only Indicate the presence of tumors but also provide detailed information about tumor types, treatment options, and severity probabilities.

While using the BraTS 2020 dataset from Kaggle. We have ensured access to a robust dataset, which is crucial for training and validating our AI model. The project also includes a well-documented website developed using HTML, CSS, JavaScript, Python, offering a comprehensive overview of the project, its methodology.

Our team consists of Saliba Rishmawi, the project leader, AI developer, and application co-developer, and Bishara AL-Babish, the web developer, and application co-developer. Together, we have combined our expertise to create an application that aims to significantly aid medical professionals in the early detection and treatment of brain tumors.

This document provides an in-depth look at the project's components, including the machine learning model, application development, and project documentation website. It also details our workflow, future work, and contributions to the field. We believe that this AI-integrated application will play a pivotal role in enhancing the accuracy and efficiency of brain tumor diagnosis, ultimately improving patient care and outcomes.

## 1.1 Purpose

This project aims to show how the AI brain tumor detection app works. It explains what the app needs, what it does, and where it falls short. By breaking it down, we want everyone involved to get a clear picture of what the system can and can’t do. This clarity helps people talk and work together better.

## 1.2 Scope

This project aims to create an app that spots brain tumors in MRI scans. It uses CNNs built with PyTorch to do this. The application will have an easy-to-use interface made with PyQt. It will show results right away giving details about the tumor like sure it is how strong it is, and more. In the future, the app will keep getting better at what it does. It will also be able to find different kinds of tumor in other parts of the body.

# Chapter Two: Implementation

## 2.1 Machine Learning Model (PyTorch)

The objective is to detect tumors in brain MRI scans using advanced machine learning techniques. This enables early and accurate diagnosis:

**Framework**: PyTorch - A widely-used open-source library for deep learning and neural network development. Its flexibility and performance and make it ideal for developing complex models like CNNs.

**Model Architecture**: A Convolutional Neural Network (CNN) is implemented, leveraging its proven efficacy in image analysis and pattern recognition tasks.

**Training Data**: MRI scan data is sourced from Kaggle which is called the BraTS dataset.

**Training Process**: The model undergoes supervised learning, where labeled data helps the CNN learn to differentiate between healthy and tumor-affected brain tissue. focusing on optimizing accuracy while minimizing false positives and negatives to ensure reliable tumor detection.

**Validation**: The model's performance is rigorously evaluated using cross-validation and test sets, ensuring its robustness and generalizability across different data samples.

**Preprocessing**: MRI images are normalized, resized to a consistent dimension, and augmented through techniques like rotation, flipping, and contrast adjustments. This preprocessing enhances the model’s ability to learn effectively from the data.

## 2.2 Project Documentation Website (HTML, CSS, JavaScript, Python)

The project documentation website is designed to comprehensively cover all aspects of the AI model, the application. It is built using HTML for structure, CSS for styling, JavaScript for interactivity, Python for the application.

The website includes several key sections:

* **Introduction**: A brief information about MRI and the brain tumor types and overall.
* **About Us**: Information about the team and collaborators
* **Demonstration**: An interactive demo of the application without the AI.
* **Contact Us Section**: Providing the team’s email for contacting them via email.

## 2.3 Workflow

The workflow involves several key stages:

1. **Data Preparation**: Collect and preprocess MRI scan data from partners, followed by preprocessed and annotating the scans to create a labeled dataset.
2. **Model Development**: Design and implement the CNNs architecture, train it on the labeled dataset, and then validate and fine-tune the model for optimal performance.
3. **Application Development**: Develop the user interface using PyQt, integrate the trained model into the application, and implement features for file upload, real-time analysis, and result display.
4. **Website Development**: Build the project documentation site using HTML, CSS, JavaScript, and Python, ensuring clear organization and ease of navigation, with regular updates to include new information and resources.
5. **Testing and Deployment**: Conduct thorough testing of the application for accuracy and user experience, deploy the standalone application with a focus on privacy.

## 2.4 User Interface (PyQt)

The objective is to develop an intuitive and efficient application for users to upload MRI scans and receive diagnostic results. The framework used is **PyQt**, which facilitates the creation of cross-platform applications that offer a native look and feel, ensuring both functionality and ease of use.

**Key Features**:

* **File Upload**: Seamless interface for users to upload MRI scan files.
* **Real-time Analysis**: Immediate processing of uploaded scans, providing on-the-spot diagnostic feedback.
* **Results Display**: Visual representation of the MRI scan with clearly highlighted areas, indicating potential tumors.
* **Tumor Information**: Comprehensive details on detected tumors, including If there is a tumor, no tumor, tumor type, and tumor locations.

## 2.5 Team Contribution

**1. Saliba Rishmawi**

**- Role:** Project Leader, AI Developer, and Application Co-Developer

**- Contributions:**

**- Project Leadership**: Directs the overall project direction, ensuring cohesive and

strategic progress.

**- AI Development:** Designed and implemented the CNN model using PyTorch.

Responsible for data preparation, model training, validation, and continuous

improvement.

**- Application Development:** Collaborated on the development of the application's

user interface using PyQt, focusing on integrating the AI model and ensuring seamless

real-time analysis of MRI scans.

**- Research and Documentation:** Contributed to the research paper and project

documentation, providing insights into the AI model and its performance.

**2. Bishara AL-Babish**

**- Role:** Web Developer and Application Co-Developer

**- Contributions:**

**- Web Development:** Developed the comprehensive project documentation website

using HTML, CSS, JavaScript, Python. Organized content to ensure clarity and

ease of navigation, including sections like about us, about the project.

**- Application Development:** Collaborated on the development of the application's

user interface using PyQt. Implemented features for file upload, real-time analysis,

and results display, ensuring a user-friendly experience.

**- Research and Documentation:** Contributed to the research paper and project

Documentation, focusing on the technical aspects of the website and the

application's user interface.

**Collaborative Efforts:**

**- Document and Research Paper:** Both team members, Saliba Rishmawi and Bishara AL-

Babish, are actively involved in drafting and refining the project documentation and

research paper. Their combined efforts ensure that all aspects of the project, from AI

development to user interface design and web documentation, are comprehensively

covered and presented.

**- Testing and Validation:** Worked together to rigorously test the application for

accuracy, usability, and robustness, gathering feedback to make necessary

improvements.

**- Future Enhancements:** Jointly planned future work and enhancements, such as

continuous model updates, feature expansion, and the development of a mobile

version of the application.

# Chapter Three: Overall Description

## 3.1 Product Perspective

* **MRI Scan Upload**: People can upload MRI scan images in DICOM format.
* **Tumor Detection**: The app uses CNNs to process the uploaded scans to spot and identify brain tumors.
* **Detailed Tumor Information**: It gives specifics about found tumors such as name, and Tumorous or not.
* **Real-Time Diagnostics**: It shows instant diagnostic results with a visual of the tumor.
* **Treatment Suggestions**: It proposes possible treatments based on the kind of tumor it finds.
* **Multi-Tumor Detection**: Future upgrades will include finding various tumor types in different body parts.

## 3.2 Product Functions

The MRI Brain Tumor Detection Application offers several powerful features aimed at improving brain tumor detection and diagnosis:

* **MRI Scan Upload:**

The app lets users upload MRI scans in DICOM format. This format is the go-to standard for medical imaging, which means the app works well with most MRI machines out there. users have options: they can drag and drop files or look through their folders to kick off the analysis.

* **Tumor Detection:**

The app uses CNNs built in PyTorch to process MRI scans and find brain tumors with great precision. This AI system spots and marks tumor areas in the MRI pictures giving a clear view for users to study further.

* **Detailed Tumor Information:**

Once a tumor is spotted, the app gives full details such as the tumor's name, a description how big it is how intense it is, error rates, and how sure it is about finding the tumor. This information is a key for users to make smart choices about patient care and to rely on it as a way to spot tumors and get the detailed information about it.

* **Real-Time Diagnostic Results:**

The app shows diagnostic findings in real time as it processes the MRI scan. Users can view the tumor's position and intensity displayed right on the MRI images giving quick insights into the patient's state.

* **Treatment Suggestions:**

The app proposes potential treatment options based on the tumor's type and severity. This feature helps plan treatments and provides links to relevant medical studies and guidelines to aid in making decisions.

* **Multi-Tumor Detection (Future Enhancements):**

In the future, updates will make the app able to spot different kinds of tumors in various parts of the body. This new feature is part of ongoing work to make the app more useful and flexible.

## 3.3 User Classes and Characteristics

The MRI Brain Tumor Recognition Application serves different patients with specific needs and skills:

* **Medical Professionals:** 
  + **Radiologists:** These users make up the main audience. They use the app to diagnose and review brain MRI scans. The tool helps them improve their diagnosis skills letting them spot tumors faster and more.
  + **Oncologists:** Cancer users can use the app to get detailed info about the tumor. This helps them plan the right treatment.
  + **Neurosurgeons:** Before surgery, brain surgeons can use the app to see the tumor and check its features. This leads to better surgery results.
* **Medical Students and Researchers:** 
  + **Students:** Medical students can use this app as a learning tool to grasp brain tumor detection complexities and AI's role in current medical practices.
  + **Researchers:** This app allows researchers to examine data, confirm their findings, and help enhance the AI model.

# Chapter Four: Specific Requirements

## 4.1 Functional Requirements

* **MRI Scan Upload:**

Users can upload MRI scans in the DICOM format through the app. The app checks the uploaded MRI scan to make sure it’s compatible and of good quality before its start processing.

* **Tumor Detection:**

The app uses CNNs built with PyTorch to spot and identify brain tumors in the uploaded MRI scans. The app gives instant updates on how the detection is going showing a picture of the found tumor on the MRI scan.

* **Detailed Tumor Information:**

The app creates a report with the tumor’s name, Tumorous or not… after detection. Doctors can view, save, and print the detailed tumor report.

* **Treatment Suggestions:**

The app suggests possible treatment options based on the detected tumor type. The app provides links or references to medical literature about the suggested treatments.

* **Multi-Tumor Detection (Future Enhancement):**

In upcoming updates, the app will be able to spot and name different kinds of tumors in various body parts.

## 4.2 Non-Functional Requirements

* **Performance:**

**Not Determined yet and will be after the AI integration into the application.**

* + s
  + s
* **Usability**:

The app will have an easy-to-use interface that users and patients can navigate without trouble. The app provides a simple way to upload MRI scans and get detection results.

**Reliability**:

The app will keep a high level of accuracy in detection, with an error rate below 5% when identifying tumors. The app will have a strong system to handle errors dealing with invalid or incompatible MRI scans.

* **Security**:

The application must encrypt all MRI scans and user data during transmission and storage. It will be accessed only by the hospital and they distribute it to the doctors at that point, so it is high-level security.

**Portability**:

The application needs to work with Windows 10/11, macOS Catalina and later.

* **Maintainability**:

The application's code should be modular and well-documented to make future updates and improvements easier. The design should allow for easy addition of new AI models or features in later versions.

**The MRI Brain Tumor Recognition Application works well on different platforms making it easy to access and use for many people:**

* **Operating Systems:**

The app runs on main operating systems such as Windows 10/11, macOS Catalina and newer versions, and Linux. This means patients can use the application no matter the system they use.

* **Hardware Requirements:**
  1. **Server-Side:**
* Web Server: A dedicated or cloud-based server capable of running a web server software and I used xampp as an option.
  1. **Client-Side:**
* Devices: The website should be accessible from desktops, laptops, tables.
* Internet Connection: Users must have an internet connection to access the website to read about MRI and using How to Demo page which demos without an AI.

**Not Determined yet and will be after the AI integration into the application.**

**To run the app without issues, users need:**

* + s
* **Software Requirements:**

**1- Server-Side Software:**

* Web Server Software: Netifly.com [Our Website](https://mridetection.netlify.app/)
* Server-Side Scripting Language: HTML, CSS, JavaScript, Python.

**2- Client-Side Software:**

* Web Browsers: The website should be compatible with modern web browsers like Google Chrome, Mozilla Firefox, Microsoft Edge.

**3- Frameworks:**

* Front-End Framework: HTML and CSS and JavaScript.

Python 3.8 or later serves as the foundation for the application, which uses PyTorch to handle deep learning tasks. PyQt creates the user interface on desktop computers. The app also has a DICOM Viewer to make it easy to see MRI scans. This viewer might be part of the app or work as a separate tool, depending on which platform you're using.

**The languages we used to make the project:**

**HTML (Hypertext Markup Language):** is the standard language for creating and structuring web pages. It uses a series of elements and tags to define the content and layout, including headings, paragraphs, links, images, and other multimedia. HTML provides the foundation for web pages, which can be styled with CSS and made interactive with JavaScript.

**JavaScript:** is a versatile programming language used primarily for creating interactive and dynamic content on websites. It runs in web browsers, enabling developers to modify HTML and CSS, handle user inputs, and perform asynchronous operations. Popular frameworks like React and Angular, and runtime environments like Node.js, extend its capabilities to mobile apps, server-side development, and more.

**CSS (Cascading Style Sheets):** is a stylesheet language used to describe the presentation of a document written in HTML or XML. It controls the layout, colors, fonts, and overall visual appearance of web pages, making them attractive and user-friendly. CSS enables responsive design, ensuring websites look good on all devices, and allows for the separation of content (HTML) from design (CSS).



**Figure 4.1: HTML,CSS,JS Logos**

**Python:** is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically type-checked and garbage-collected.

* **Network Requirements:**
  + You need an internet connection to access the website online.
  + the main detection and diagnostic features work offline. This makes the application reliable and easy to use even when you're not connected to the internet.

## 4.3 Interface Requirements

* **User Interface (UI)**:

The app will have a GUI. PyQt will be used to design the desktop version. users can access and understand these elements.

* **Data Interface**:

The app will work with DICOM format for MRI scans. This ensures it's compatible with standard medical imaging devices. users can export tumor reports as PDFs. This makes sharing and printing easy.

* **External System Interfaces**:

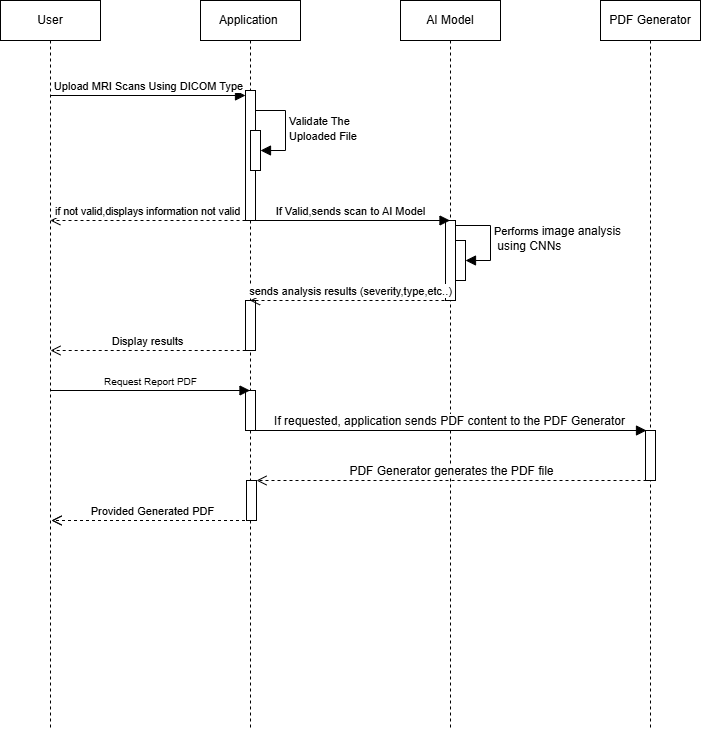
The app will work with outside DICOM viewers, so patients can look at MRI scans in depth elsewhere if they need to. The app will also be able to link up with online medical databases to get the newest treatment choices and related medical info.

* **Error Handling Interface**:

When users put in wrong info fail to upload scans, or the app can't detect something, it will show clear short error messages. These messages will tell users how to fix the problem or where to go for more help.

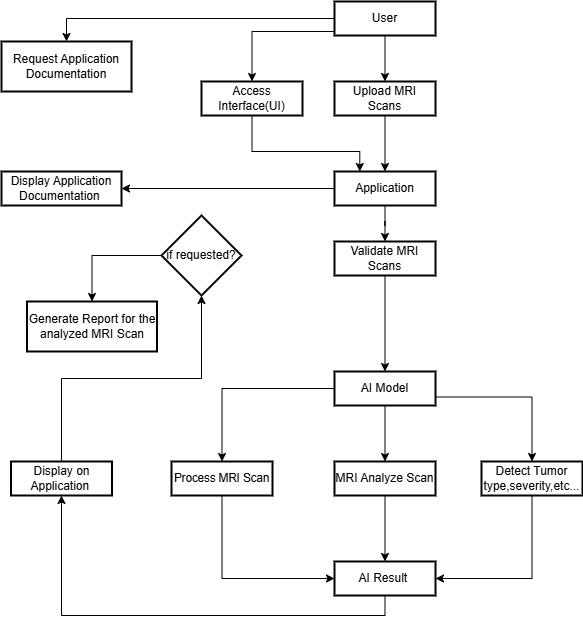
# Chapter Five: Diagrams

## 5.1 Sequence Diagram



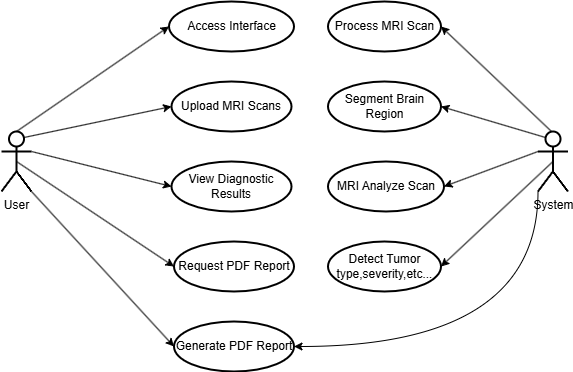
**Figure 5.3: Sequence Diagram**

## 5.2 Data Flow Diagram



**Figure 5.4: Data Flow Diagram**

## 5.3 User Case Diagram



**Figure 5.5: Use Case Diagram**

## 5.4 Entity Relationship Diagram

## 5.5 Class Diagram

# Chapter Six: Breakdown

# Chapter Seven: Prototype

**Screenshots from the prototype are included below to demonstrate core features.**

**Each is accompanied with a description.**

## 7.1 Header

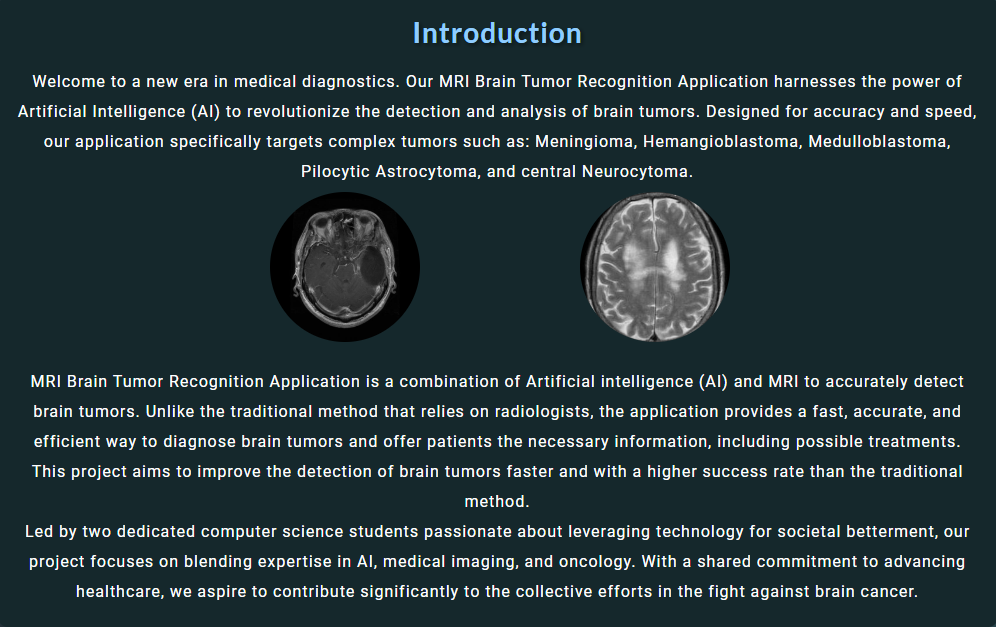




**Figure 7.1: Header of the Website**

**The website’s main header has a hover menu with buttons for** "**Introduction**"**,** "**How To Demo**"**, and** "**About us**"**. Each button leads to its own page that explains the website’s purpose and other details provided. For instance, the Introduction page talks about what MRI is and explains common knowledge, while the How To Demo page shows how the application works without using AI. It lets you choose a picture from a provided folder and displays information about it. The About Us page introduces the team behind this project, Bishara Babish and Saliba Rishmawi, with some general knowledge and background about them. Also, there’s a language switch button on the top right that changes the website’s language between English (EN) and Arabic (AR), and vice versa.**

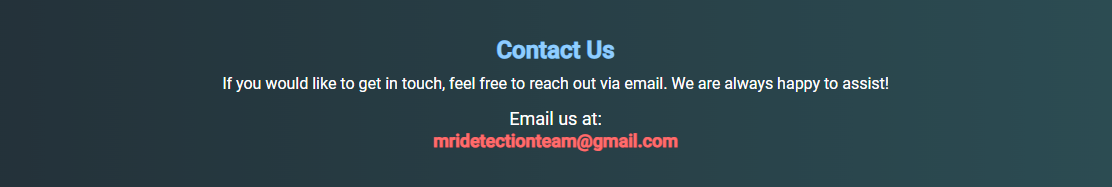
## 7.2 Introduction & Contact Us





**Figure 7.2: Introduction of the Website**

The Introduction page gives an overview of what MRI is, explaining its purpose and how it relates to our project. The main goal of the project is to use AI to identify brain tumor, specifically Glioma, No Tumor. It also discusses the application we’re developing, which detects brain tumor using image recognition. Then, it introduces the two individuals behind the project, Bishara Babish and Saliba Rishmawi both computer Science students who are focusing on AI for this project. Our Hope is to contribute meaningfully to the collective efforts in the fight against brain cancer. Also, as it shows in the second picture that it is in Arabic which we use the language switch button to change the language as provided.

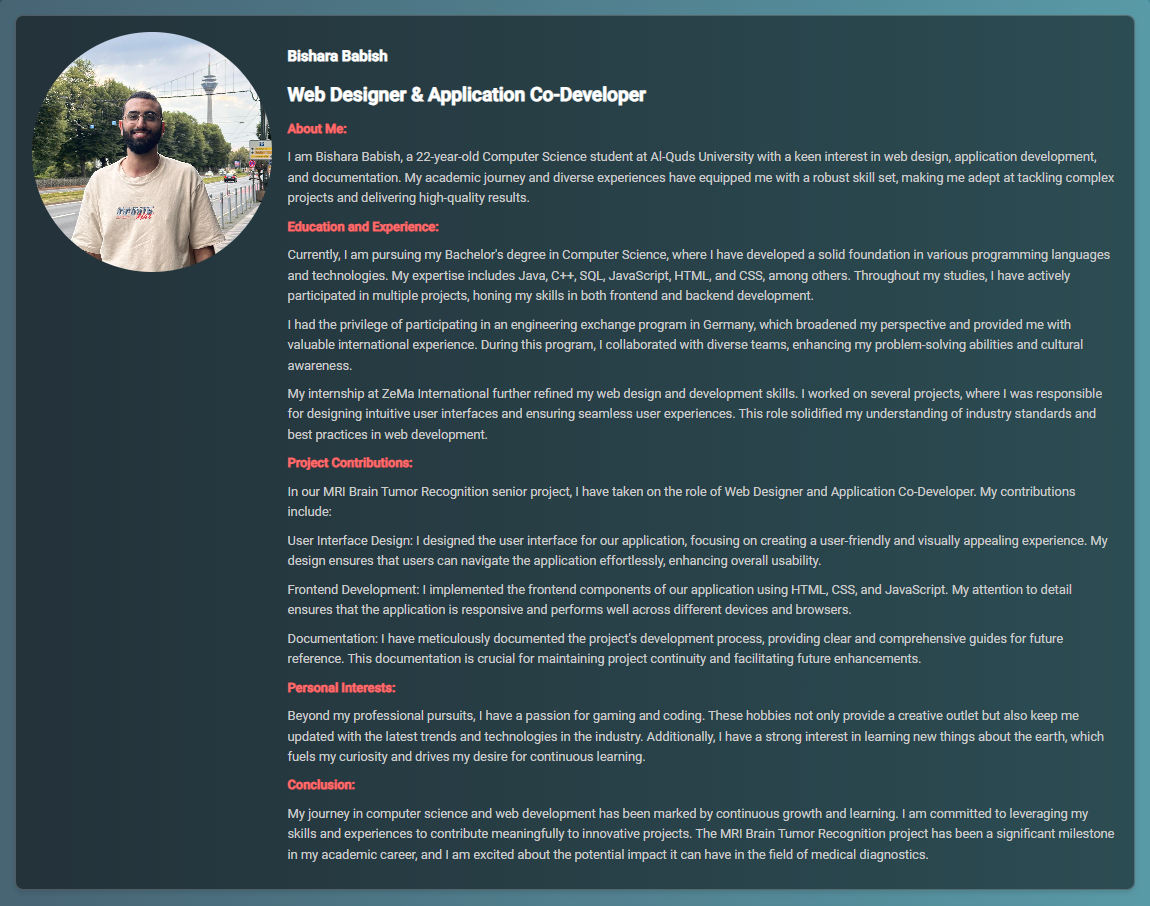




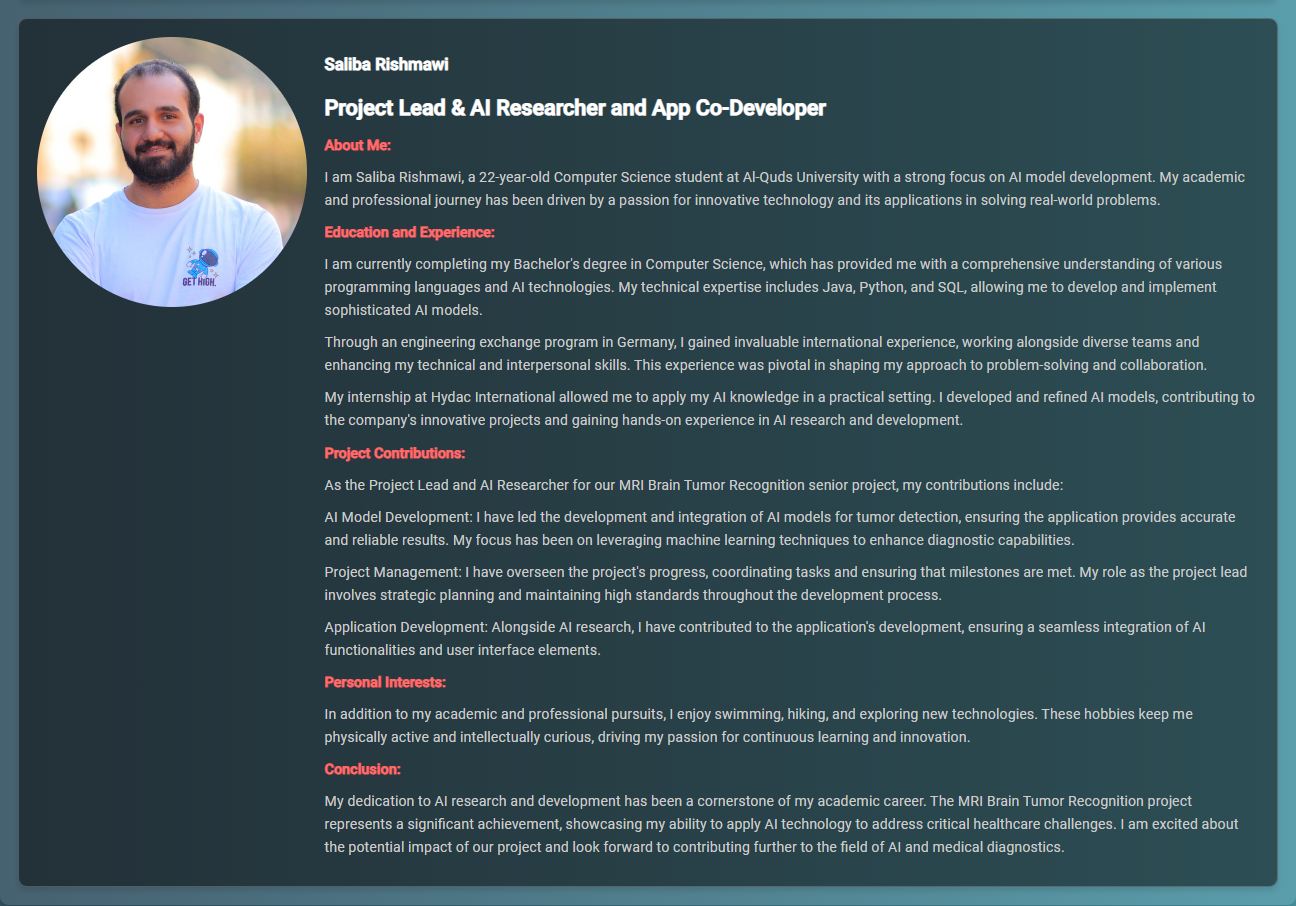
**Figure 7.3: Contact Us of the Website**

The Contact Us section shows in both Arabic and English translations, which shows the email of the Team’s email to be able to contact them via email as a hyperlink.

## 7.3 About Us



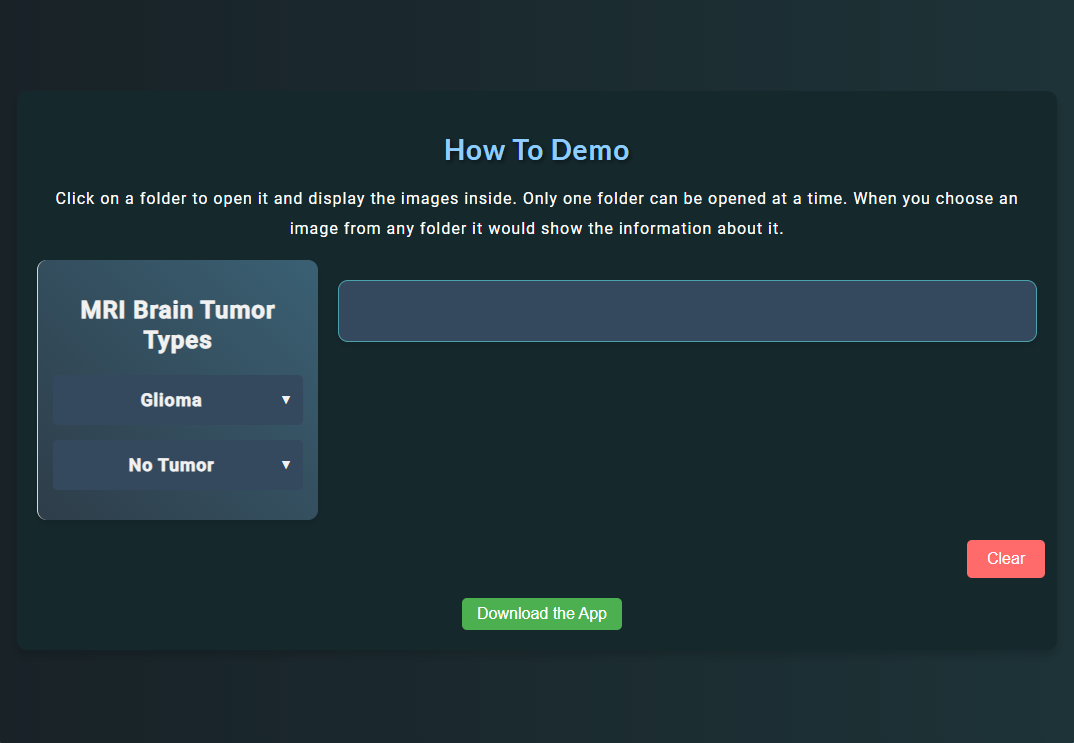


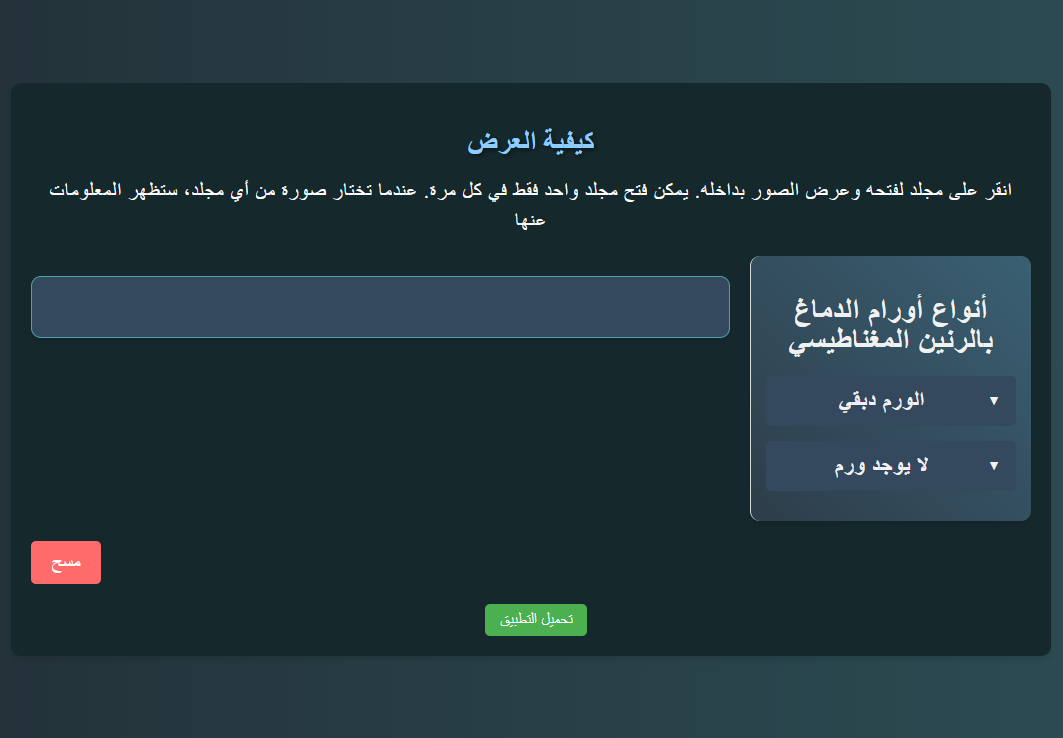




**Figure 7.4: Bishara & Saliba Overview and Information**

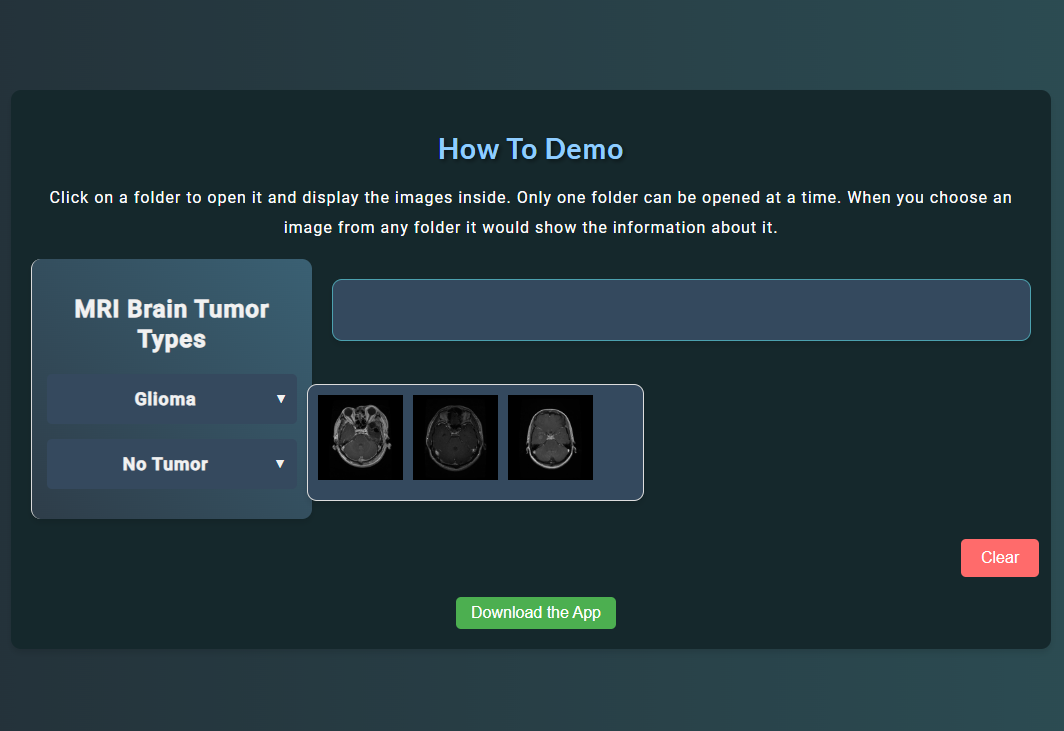
The pictures provide information about Bishara Babish and Saliba Rishmawi, covering details about their university, age, educational journey, and experiences. It also highlights the programming languages they know, their goals, and their contributions to the project. The roles they play are different, with Bishara focusing on the user interface, frontend development, and documentation, while Saliba serves as the project lead, AI researcher, and AI model developer, also handling project management and application development. This information is available in both Arabic and English. After that, it touches on their personal interests, concluding with a summary of their roles and contributions.



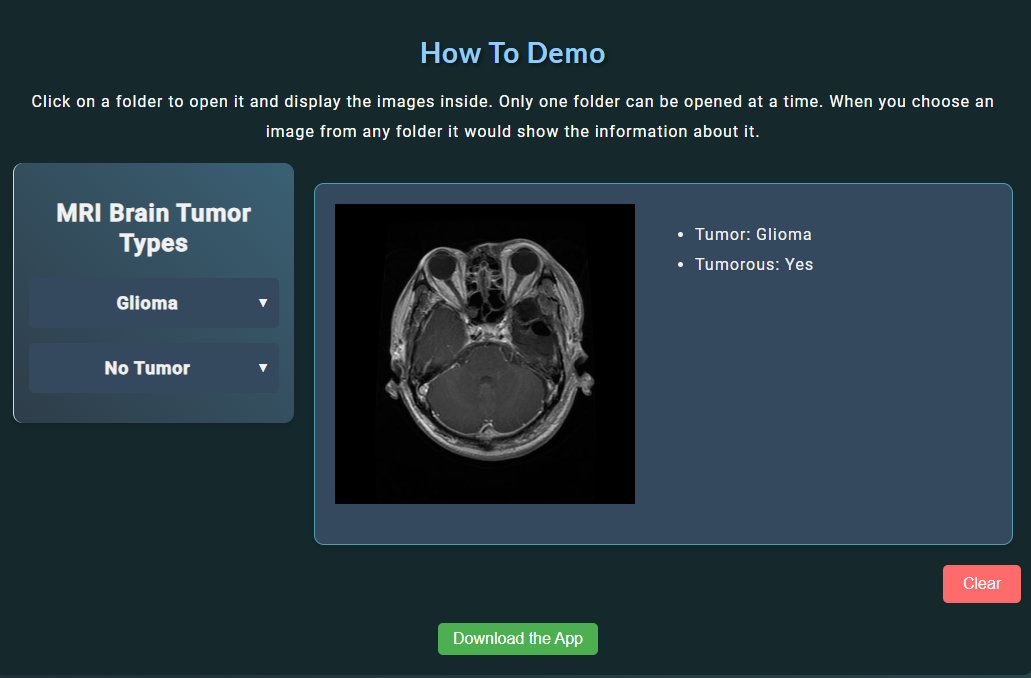


**Figure 7.5: How to Demo Demonstration**

The provided section discusses the "How To Demo" demonstration, which shows how the application works without AI. And which it includes the button “Download the App” which downloads the application that is connected with the Ai to detect the tumours. It involves selecting built-in images from a dataset provided from Kaggle which is called the BraTS. In this demo, you can browse through folders, and when you click on one, it displays detailed information, as explained in Figure 7.8.



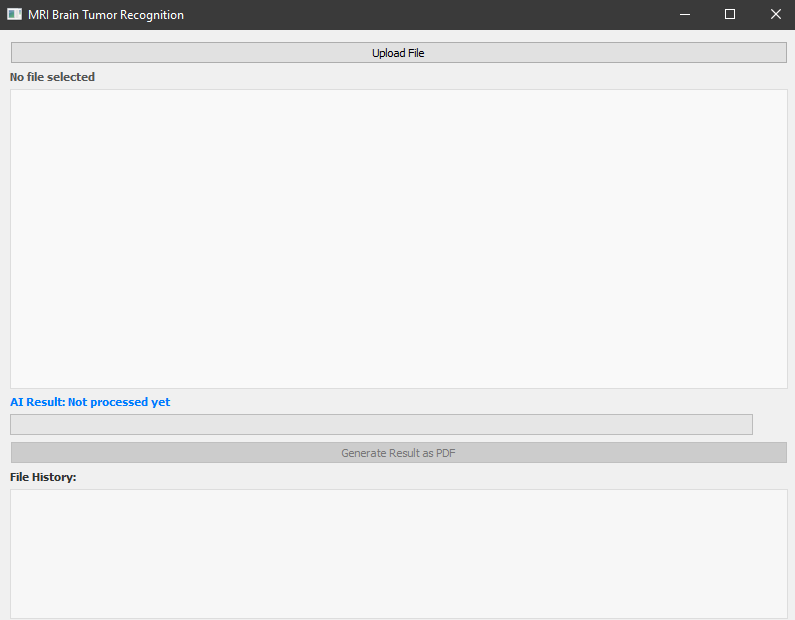






**Figure 7.6: Choosing a Picture from the folder**

This section demonstrates that when you click on a folder containing the types of tumours we're working with, it shows three different images, each with distinct information and diagnoses provided BraTS. The information is displayed in both Arabic and English, as it's built into the system. There's also a "Clear" button, which removes the selected images and the related information, allowing users to reset the view.



**Figure 7.7: Detection Application**

The provided pictures shows how the application for the MRI detection looks like which has an File History of what the patient or a doctor uploads via Upload file or drag and drop which can be downloaded on the How To Demo page Following the “Download the App” Button.

# Chapter Eight: Results and Discussion

## 8.1 Survey Introduction

To gather valuable feedback for the development of our AI-integrated brain tumor detection application, we conducted a survey targeting potential users, including medical professionals, researchers, developers, and students. The survey aimed to understand user needs, preferences, and concerns regarding the application features, user interface, and data privacy.

The survey was conducted using Google Forms and can be accessed through the following link: [Survey Form](https://docs.google.com/forms/d/e/1FAIpQLSc9_sEHu_uCqJWeqhRS7kSfq0d0wL1O2T500y57-CF5ClDT6w/viewform)

## 8.2 Survey Questions

### 8.2.1 General Information

1. What is your age?

2. What is your gender?

3. Field of study.

4. Year of study

### 8.2.2 Experience with AI Applications

1. What is your experience using AI?

2. If yes, please describe your experience with AI.

3. Do you have any experience working with MRI scans?

4. if yes, please describe your experience with MRI.

### 8.2.3 User Interface and Experience:

1. How would you prefer to upload MRI scans to the application?

2. Would you prefer for the application to generate PDF of the results after making the

analysis?

### 8.2.4 Data Privacy and Security:

1. Rank the following features from most important to least (Tumor Detection, Tumor

type identification, Severity assessment).

2. How concerned are you about data privacy when using medical applications?

### 8.2.5 Feedback and Suggestions:

1. Are there any additional features that you would like to see in the application?

2. Please specify what data concerns you have?

3. Do you have suggestions on how to improve the user experience when using the

application?

## 8.3 Survey Analysis

The survey responses were collected and analyzed to gain insights into user preferences and requirements. Below are the key findings from the survey:

### 8.3.1 General Information:

### 8.3.2 Experience with AI Applications:

### 8.3.3 User Interface and Experience:

### 8.3.4 Data Privacy and Security:

### 8.3.5 Feedback and Suggestions:

# Chapter Nine: Future Work and Enhancements

Future work includes improving the model by incorporating new data to enhance accuracy, expanding features to include detailed diagnostic reports and integration with other medical databases, and providing training materials and workshops to help users effectively utilize the application.

This detailed project description outlines the core components, technologies, workflow, and future directions of the AI-integrated brain tumor detection application. It provides a comprehensive understanding of the project's scope and objectives, ensuring clarity and focus for all stakeholders.

## 9.1 Additional Considerations

Key aspects include ensuring data security to protect user information, incorporating user feedback to improve the application, and maintaining ongoing collaboration with partners and stakeholders to support continuous development and refinement.

# Chapter Tenth: References

* MRI Brain Tumor Recognition Using PyTorch and CNNs. PyTorch Documentation. Available at: <https://pytorch.org>
* PyQt Documentation. Available at: <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
* DICOM Standard. Available at: <https://www.dicomstandard.org/>
* Radiology Imaging Techniques. Radiology Society of North America. Available at: <https://www.rsna.org/>
* Machine Learning in Medical Imaging. Available at: <https://www.ncbi.nlm.nih.gov/pmc/>
* HTML, CSS, and JavaScript Guide. Mozilla Developer Network. Available at: <https://developer.mozilla.org/>

# Chapter Eleventh: Conclusion

The survey results provide valuable insights that will guide the development of the AI-integrated brain tumor detection application. By incorporating user feedback, we aim to create a tool that meets the needs of medical professionals and researchers, ensuring accuracy, usability, and data security. The positive response and engagement from potential users highlight the importance and potential impact of this project.

This survey and analysis section will be continuously updated as we receive more feedback and iterate on the application's design and functionality.