**CHAPTER 1**

**INTRODUCTION**

1. **INTRODUCTION**
   1. **Problem Summary**

Alzheimer's disease (AD) is a degenerative neurological disorder that causes short-term memory loss, psychosis, and delusional thoughts that are misdiagnosed as signs of stress or ageing. About 5.1 million Americans are afflicted by this disease. There is inadequate medical care for AD.

A consistent medication schedule is required to manage AD. Since AD is chronic, it might last for a long time or for the rest of your life. Therefore, in order to prevent significant brain damage, it is crucial to prescribe medication at the right time. Early diagnosis of this disease is a time-consuming and expensive process since we need to gather a lot of data, apply advanced algorithms for prediction, and include an expert physician. Because automated systems are immune to human mistake, they can be employed in medical decision support systems and are more accurate than human assessment.

The use of pictures (MRI scans), biomarkers, and other techniques has been applied to prior studies on AD. To explore this disease, researchers used (chemicals, blood flow), as well as numerical data taken from MRI scans.

As a result, they could determine whether or not a person was insane. Automating Alzheimer's diagnosis will decrease further human interaction in addition to cutting down on diagnosis time. Automation also lowers overall expenses and yields more precise results.

For example, we can predict whether a patient is demented by analysing MRI scans and applying prediction techniques. If a person has early-stage Alzheimer’s Disease, they are considered demented. By doing so, we can achieve better accuracy. When a person has Alzheimer’s Disease in the early stages, they can usually function without any assistance. In some cases, the person can still work, drive, and partake in social activities.

* 1. **Project Purpose**

Imagine an app that analyses your risk for Alzheimer's disease. By considering factor like Human’s Brain MRI scan, it could identify individuals at higher risk. This wouldn't be a diagnosis, but a prompt for further evaluation. Early detection is crucial, and such an app could empower people to seek a doctor's assessment sooner. This could lead to earlier intervention with treatment or lifestyle changes, potentially improving outcomes.

Healthcare professionals could also benefit from this technology in identifying at-risk patients. While still under development, this application holds promise for the future of Alzheimer's disease management.

* 1. **Objectives**
* 24-hour service
* Getting an instant response
* Friendly and easy to approach
* Secure
* Less Time Consuming
  1. **Project future Scope**
* **Advanced Risk Assessment:** Go beyond traditional factors to include cognitive tests, wearables, and potentially genetics.
* **AI-powered Accuracy:** Leverage machine learning for more precise risk prediction.
* **Personalized Action Plans:** Provide tailored recommendations for diet, exercise, and potential clinical trials.
* **Predictive Tracking:** Track potential disease progression for better disease management.
* **Telehealth Integration:** Connect users with specialists for remote monitoring and improved access to care.
  1. **Project Specification**

The Alzheimer's disease prediction application project aspires to create a mobile app that surpasses a basic risk assessment tool. It will delve deeper by analysing user data on established risk factors like Brain MRI. Additionally, the app has the potential to integrate with wearable sensors and genetic testing services (with user consent) to paint a more comprehensive picture of an individual's risk. Machine learning algorithms will continuously refine risk prediction accuracy, while personalized recommendations will go beyond basic advice. Users can expect tailored suggestions for lifestyle modifications and potentially even be matched with relevant clinical trials (future development). The app's future advancements could encompass tracking disease progression and seamlessly connecting users with specialists via telehealth services. To empower individuals and promote informed decision-making, the app will incorporate educational resources about Alzheimer's disease. Transparency regarding the app's limitations and the importance of professional evaluation will always be emphasized. This project strives to be a multifaceted approach to early detection and potential management of Alzheimer's disease, ultimately empowering individuals to take charge of their brain health.

* 1. **Technology**
* **Frontend**
* Dart
* **Backend**
* Python
* Flask-API
* TensorFlow
* Scikit-learn

**CHAPTER 2**

**LITERATURE REVIEW**

1. **LITERATURE REVIEW**
   1. **Literature review**

In initial research, I referred to some of the research paper published by authors, some

Of them are: -

1. **Burns:** Alzheimer's disease is generally considered a classical example of pathological cognitive aging (Burns et al., 1990; P. Baltes & M. Baltes, 1990). Neuropathological studies of the aging brain have primarily focussed on the loss of neurons, a reduction in size of neuronal populations, and a decrease in overall size within the aging brain.
2. **Emad Essam Shoukry M.D:** Most of the patients with Alzheimer’s disease (AD) are found to have evidence of atherosclerotic cerebrovascular disease on autopsy. Cardiovascular risk factors including hypertension, diabetes mellitus and dyslipidaemia are known risk factors for Alzheimer’s disease as well.
3. **Evelien Nackaerts:** Patients with Parkinson’s disease (PD) suffer from severe motor symptoms which can only be partly alleviated by means of dopaminergic medication. Motor rehabilitation, i.e. relearning of a known motor skill through intensive practice, can be an effective and lasting therapeutic supplement in chronic neurodegenerative diseases.
   1. **Proposed system**

An Alzheimer's disease prediction system could use machine learning algorithms to analyse a combination of factors. This might include genetic data, brain scans, cognitive tests, and even factors like age and questionnaire. By analysing this data, the system could identify patterns that increase the risk of Alzheimer's, potentially allowing for earlier intervention and treatment.

**CHAPTER 3**

**SOFTWARE REQUIREMENTS**

**SPECIFICATION**

1. **SOFTWARE REQUIREMENT SPECIFICATION**
   1. **Software Requirements: -**

* FRONT END : Dart
* BACKEND TOOL : Python, Fast-API, TensorFlow, Scikit-Learn
* OPERATING SYSYTEM : Windows 7 & above
  1. **Hardware Requirements: -**
* RAM : Minimum 1GB – Maximum Any
* ROM : 500GB
* PROCESSOR : Core 2 Duo 2.80 GHz or Higher
* DISK SPACE : 2-3 GB
  1. **Application Environment: -**
* Any browser
* Mobile Compatible
* Tablet Compatible
* Authentication

**CHAPTER 4**

**PROJECT**

**MANAGEMENT**

1. **PROJECT MANAGEMENT**
   1. **Project Development Approach:**

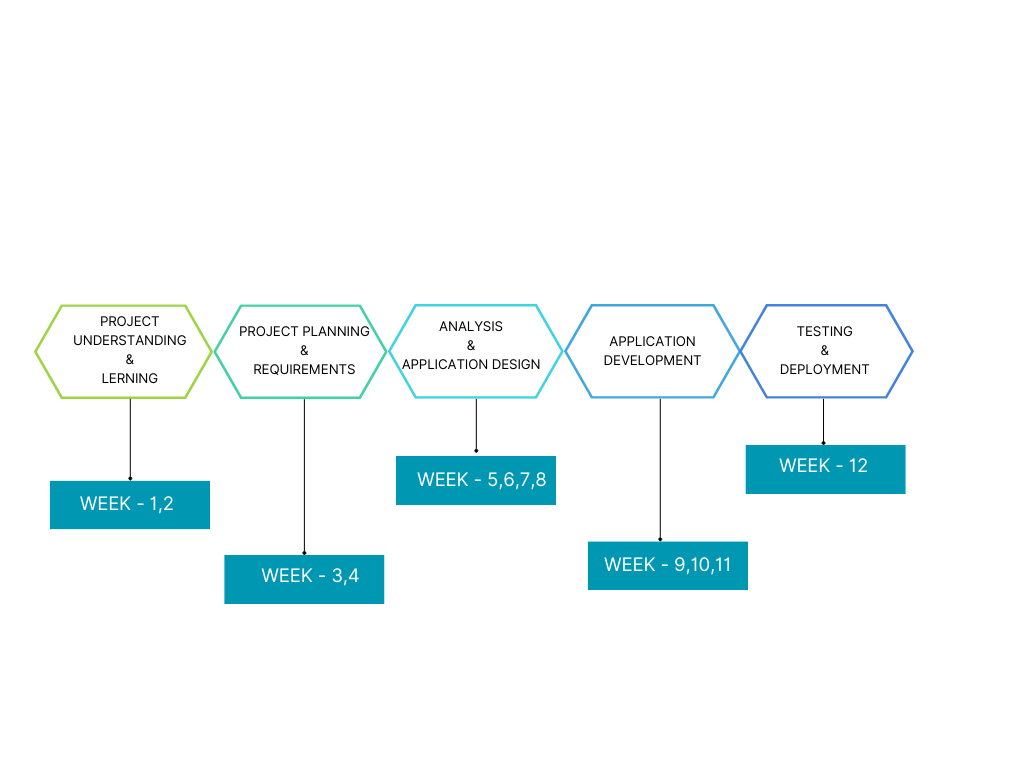
A Software Process model is simplified abstract representation of a software process, which is presented from a particular perspective. Planning prepares a framework that makes a reasonable estimate of the project. To accomplish it, software development models are used. Incremental model is used to.

* 1. **Project Planning:**

Project planning includes description of the project task, activity and function, dependencies, resources requirements and detail schedules. Project planning involves estimating how much time, efforts, money and resources will be required to build specific software system.

* 1. **Project Scheduling:**

Our Project scheduling for the Alzheimer’s Disease Diagnosis Application involves several key steps. First, I define clear project objectives and scope, outlining the features and functionalities required. Next, I create a detailed work breakdown structure (WBS) to identify all tasks and subtasks, including design, development, data integration, testing, and deployment. We then estimate task durations, allocate resources, and establish dependencies to create a realistic project timeline. Regular monitoring and adjustments are made to the schedule as needed to ensure milestones are met and project success.



**[**Fig. 4.3 – Project Scheduling]

* 1. **Risk Management:**

Risk management for a price comparison website involves identifying potential threats and developing strategies to mitigate them. Common risks include technical issues like server downtime, data breaches, or compatibility issues with different devices and browsers.

To address these risks, measures such as implementing robust security protocols, regular backups of data, and contingency plans for technical failures are put in place.

Additionally, risks related to market competition, changes in user behaviour, or regulatory requirements are considered and strategies are developed to adapt to such changes effectively.

Regular monitoring and evaluation of risks are conducted throughout the project to ensure timely detection and mitigation of any emerging threats.

* + 1. **Risk Identification:**

Identifying risks for a price comparison website involves recognizing potential issues that could impact its success. These risks may include technical challenges like server crashes, data security breaches, or application downtime.

Other risks could stem from market competition, changes in user preferences, or legal and regulatory compliance issues. To identify these risks, thorough analysis and assessments are conducted at different stages of the project.

This includes reviewing the technology stack, evaluating market trends, and understanding user expectations and industry standards. By identifying these risks early on, appropriate measures can be taken to mitigate them and ensure the smooth operation of the price comparison website.

* + 1. **Risk Analysis:**

All of the risk analysis activities presented to point have a single goal – to assist the project team in developing a strategy for dealing with risk. An effective strategy must consider three issues:

1. Risk avoidance
2. Risk monitoring
3. Risk Management and contingency planning
   * 1. **Risk Planning:**

* **Risk Mitigation Strategies:** Develop strategies to mitigate or reduce the likelihood and impact of identified risks. This may include implementing redundancy measures, conducting thorough testing, or procuring backup resources.
* **Contingency Planning:** Prepare contingency plans to address risks that cannot be fully mitigated. Define alternative courses of action and response protocols to minimize the impact of adverse events on project outcomes.
* **Risk Monitoring and Control:** Establish procedures for ongoing risk monitoring and control throughout the project lifecycle. Regularly review and update the risk register, track risk indicators, and implement corrective actions as needed.
* **Communication Plan:** Define a communication plan to ensure stakeholders are informed about potential risks, mitigation efforts, and contingency plans. Maintain open channels of communication to facilitate timely decision-making and risk response.

**CHAPTER 5**

**SYSTEM ANALYSIS**

**&**

**DESIGN**

1. **SYSTEM ANALYSIS & DESIGN**
   1. **Requirement of new system:**

* Image Preprocessing Automation
* Explainable AI (XAI)
* Multi-disease detection
* Easy Image Uploading
* Prediction Display
* Recommendations Details
  1. **Feasibility Study:**

This feasibility study assesses the viability of developing a Alzheiemer disease prediction system. Here's an analysis of key factors:

Four types of feasibility:

1. Economic Feasibility
2. Technical Feasibility
3. Operational Feasibility
4. Implementation Feasibility
   * 1. **Economic Feasibility**

This assesses if a project makes financial sense. It involves a cost-benefit analysis, considering startup costs, ongoing expenses, potential revenue, and return on investment. Essentially, will it bring in more money than it costs?

* + 1. **Technical Feasibility**

This examines if the project can be built or completed with current technology and expertise. It looks at things like the availability of the necessary resources, skills, and technologies to make the project a reality. Can it be done, given the current state of affairs?

* + 1. **Operational Feasibility**

This analyses if the project can be practically implemented within the existing organizational structure and day-to-day operations. It considers workforce needs, management style, and business processes to see if they can adapt to the project's demands. Can it fit into the way things are currently done?

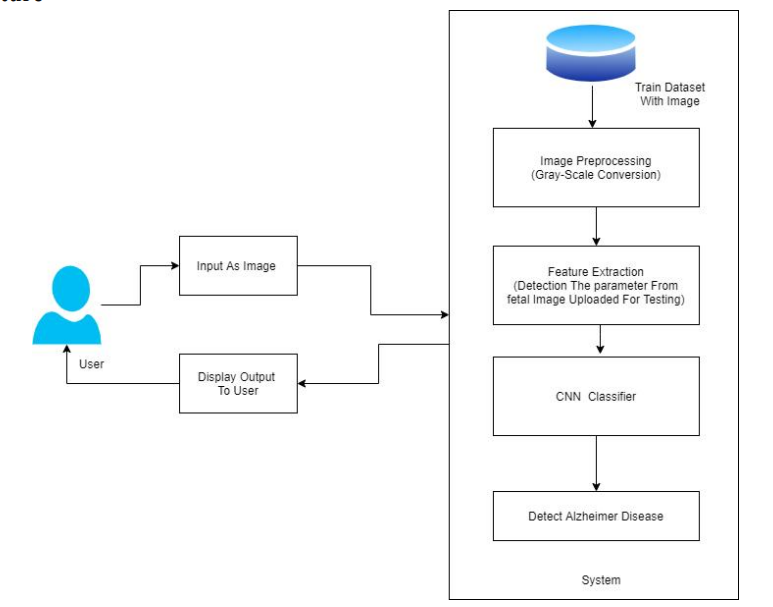
* + 1. **Implementation Feasibility**

This dives into the logistics of getting the project off the ground. It involves creating a plan that outlines the steps needed, the resources required, and the potential challenges that might arise during execution. Is there a clear roadmap to make it happen?

* 1. **Requirement Validation:**

All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be Using the system) expects from the system. The requirements are gathered from the end- user by consultation, these requirements are analysed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification Document is created which serves the purpose.

* 1. **Feature of New System:**
* Search functionality
* Multilingual Support
* Responsive Design
* Data Storage
* Less time consuming
* Easy to use
  1. **Data flow and Modelling:**
     1. **Flow Chart:**



[Fig. 5.5.1 – Flow Chart]

* + 1. **Use Case Diagram:**

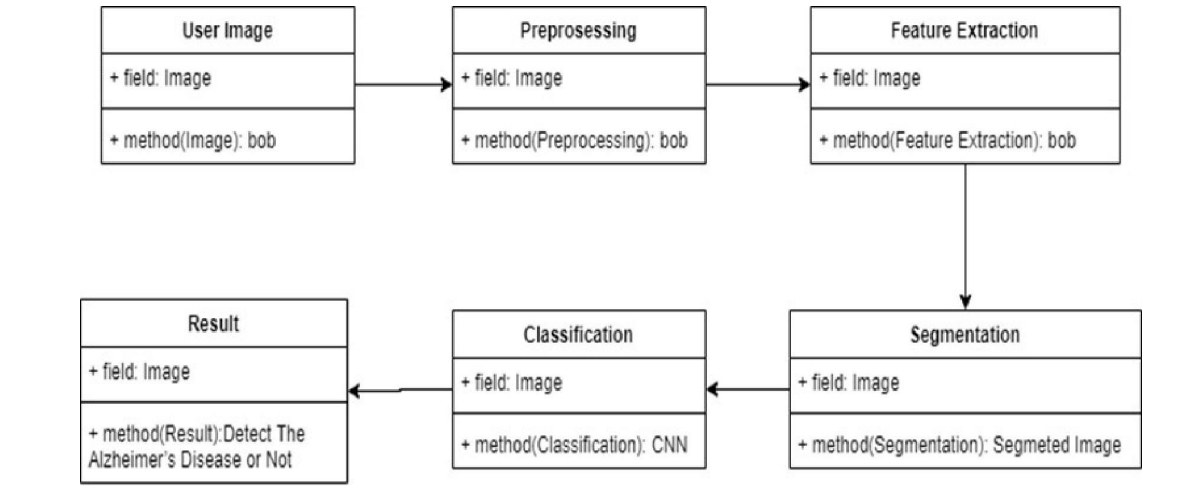
A Use Case diagram is a visual representation of how users interact with a system. It shows the various actions (use cases) user can perform and how these actions relate to each other and to the system itself.



[Fig. 5.5.2 – Use Case diagram]

* + 1. **Class Diagram:**

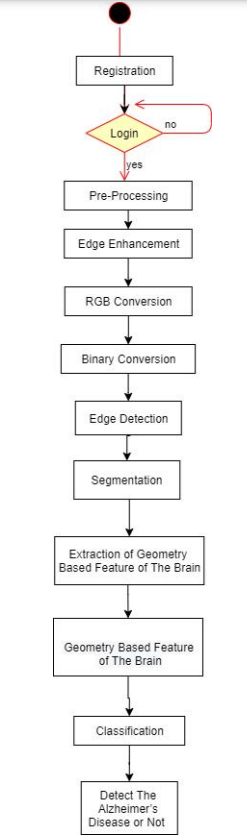
It shows the building blocks (classes) and their properties (attributes), the actions they can take (methods), and how they all work together (relationships). This helps developers, analysts, and stakeholders understand the system's structure, communicate effectively, and ultimately build a well-designed software system.



[Fig. 5.5.3 – Class Diagram]

* + 1. **Activity Diagram:**

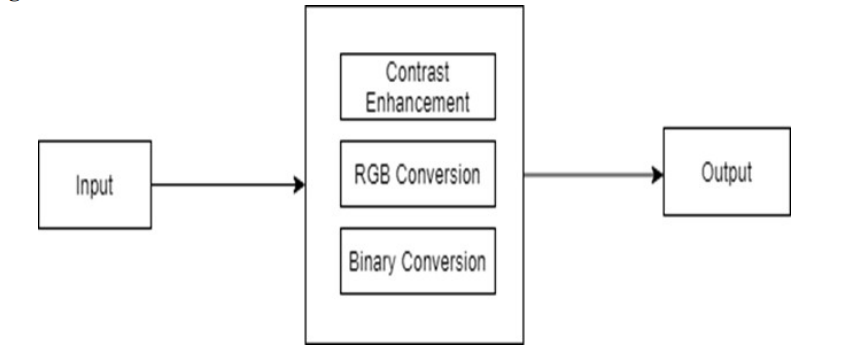
An activity diagram is a flowchart-like illustration within UML that depicts the step-by-step workflow of a process, including decision points and potential parallel activities.

****

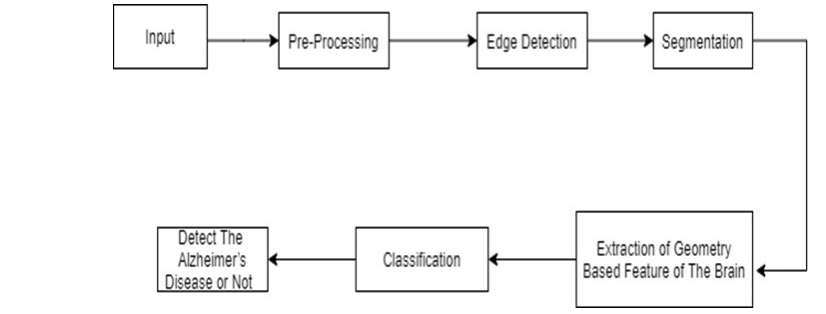
[Fig. 5.5.6 – Activity Diagram]

* + 1. **DFD:**

DFD stands for **Data Flow Diagram**. It's a graphical tool used to visualize the flow of information through a system or process. It helps us understand how data moves around, what transformations it undergoes, and how it interacts with different parts of the system.



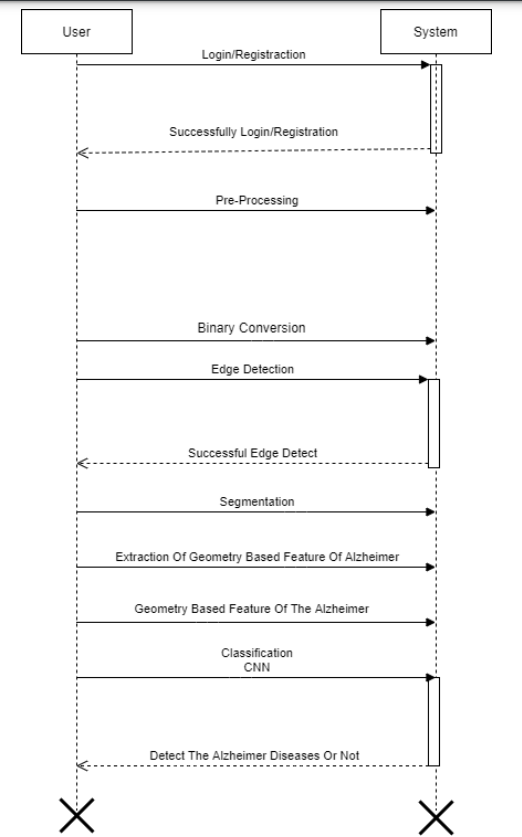
[Fig. 5.5.5.1 – DFD Level 0]

****

[Fig. 5.5.5.2 – DFD Level 1]

* + 1. **Sequence Diagram:**

A sequence diagram is a type of interaction diagram used in Unified Modelling Language (UML) to visualize the flow of messages between objects in a system. It focuses on the interactions happening over time, depicting how objects collaborate to achieve a specific functionality.

****

[Fig. 5.5.8 – Sequence Diagram]

**CHAPTER 6**

**DATA DICTIONARY**

1. **DATA DICTIONARY**
   1. **WHAT IS DATA DICTIONARY?**

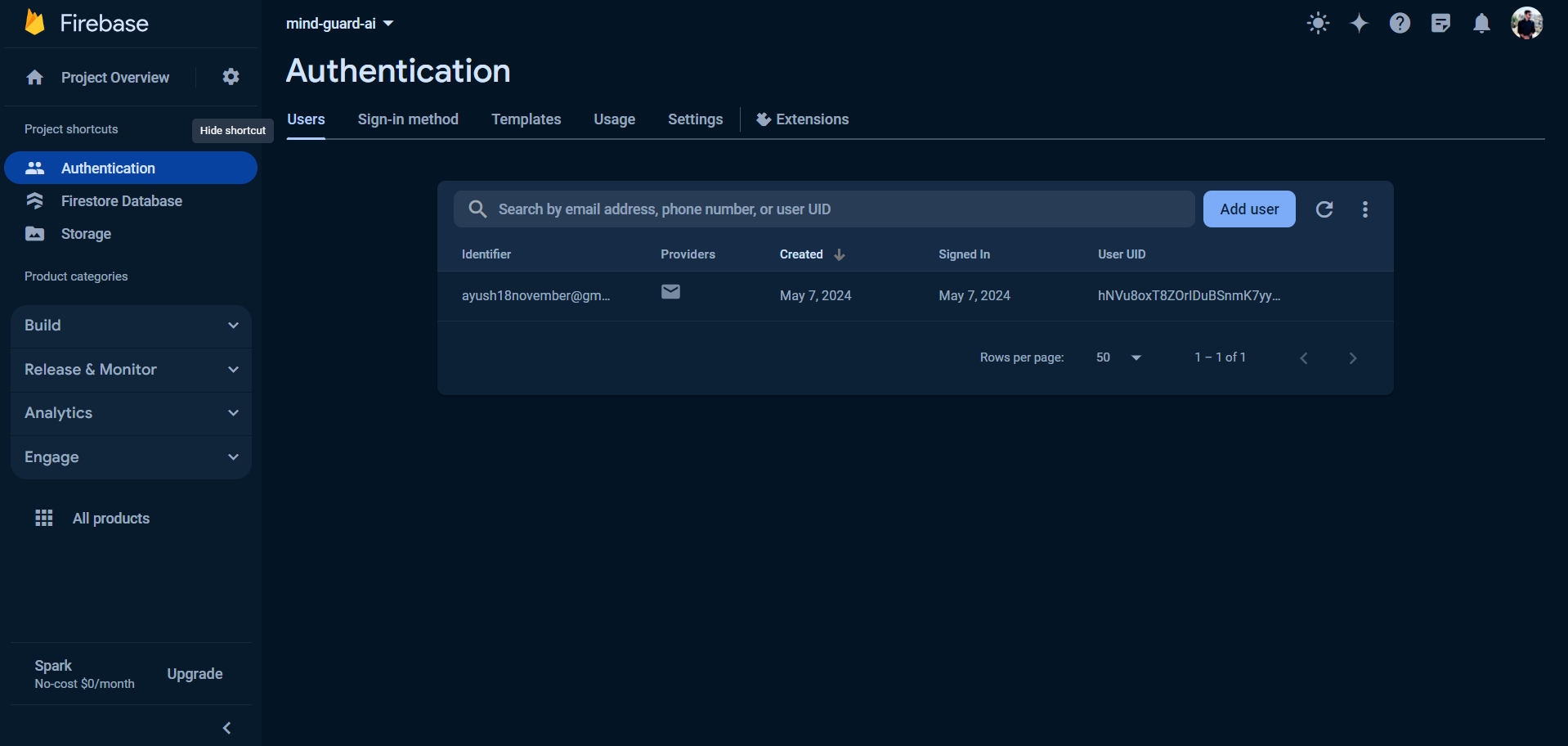
A data dictionary is a structured repository or document that provides detailed information about the data elements used in a database, software application, or project. It serves as a reference guide for understanding the structure, semantics, and usage of data within the system.

**Key components of a data dictionary typically include:**

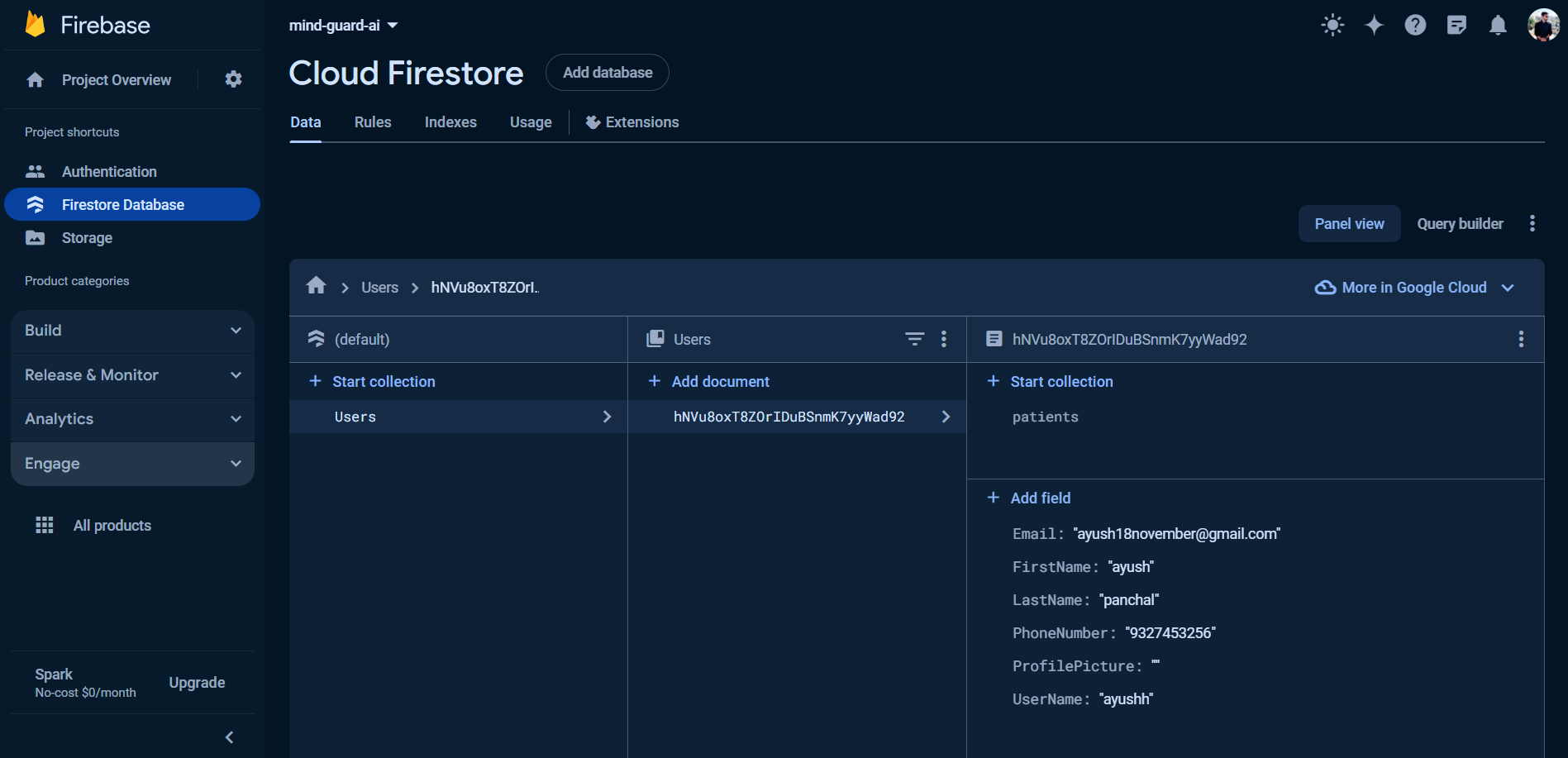
* + - Data Element Name: The name or identifier of each data element.
    - Description: A brief description or definition of the data element, explaining its purpose and meaning.
    - Data Type: The type or format of the data (e.g., integer, string, date).
    - Length: The maximum length or size of the data element.
    - Allowed Values: The permissible values or range of values for the data element.
    - Relationships: Information about relationships between data elements, such as primary keys, foreign keys, and dependencies.
    - Constraints: Any constraints or rules that apply to the data element, such as uniqueness constraints or required fields.
    - Usage: Information about how the data element is used within the system or application.
    - Metadata: Additional metadata about the data element, such as creation date, author, or modification history.

Overall, a data dictionary serves as a comprehensive reference resource for developers, database administrators, and other stakeholders involved in the design, implementation, and maintenance of a system or database. It promotes consistency, clarity, and understanding of the data structure and facilitates effective communication and collaboration among project team members.

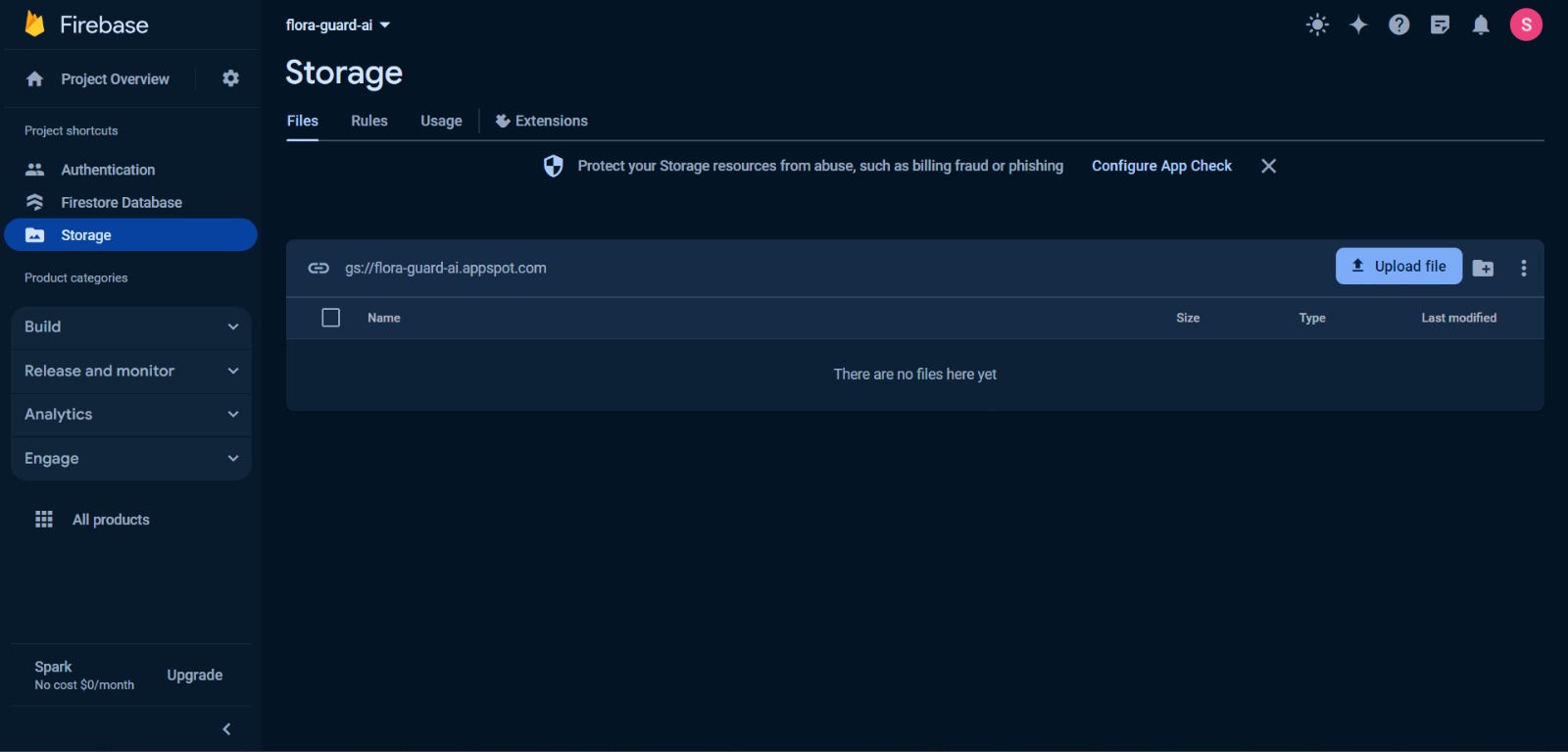
* 1. **DATA DICTIONARY**

****

[Fig. 6.2.1 - Authentication]

****

[Fig. 6.2.2 – Cloud Firestore]

****

[Fig. 6.2.3 - Storage]

**CHAPTER 7**

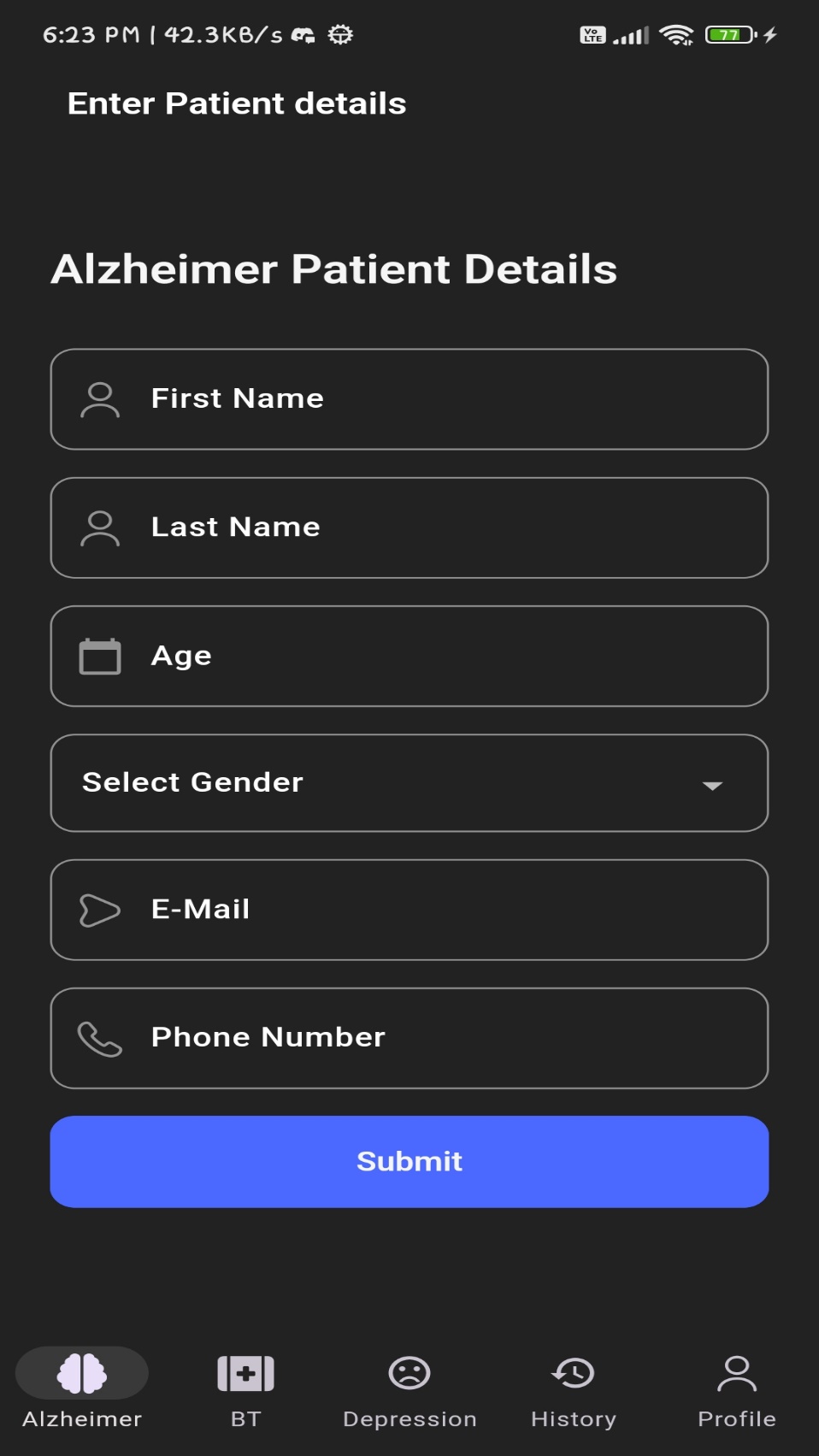
**IMPLEMENTATION**

1. **IMPLEMENTATION**
   1. **FRONTEND IMPLEMENTATION**

In a Alzheimer disease detection project using machine learning and computer vision, several tools and libraries are commonly used to design frontend (User Interface) of the model. Some of the key tools and libraries used in such projects include:

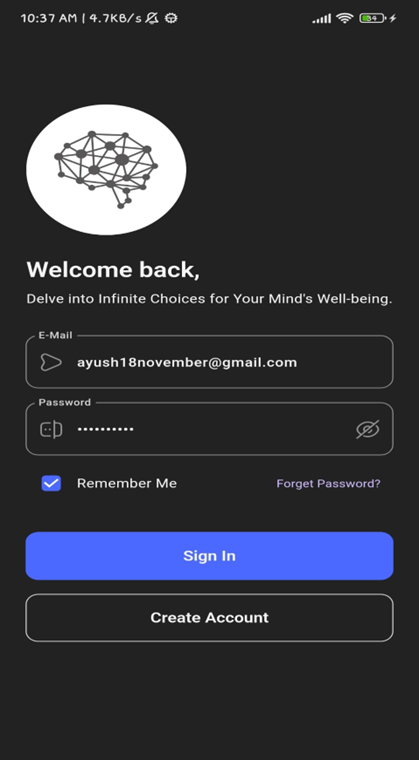
* **Flutter**
* Flutter is a mobile app SDK (software development kit) for building high-performance, high-fidelity apps for iOS and Android. With powerful graphics and animation libraries, the Flutter framework makes it easy to build use interfaces that react smoothly in response to touch.
* Flutter uses its own rendering engine to provide a smooth and native look and feel on different devices. It comes with a rich set of widgets that make building attractive and user-friendly interfaces easier.
* **Dart**
* Dart is a free, Open-source programming language that can be used to develop apps for any platform. It’s known for being approachable, productive, and portable. Dart has familiar syntax, and includes features like hot reload, profiling, logging, and debugging. It’s also fast, with just-in-time compilation and ahead of time compilation.
* Dart isn't directly used for general Android development, its integration with Flutter unlocks several advantages: faster development with hot reload, code reusability across platforms, performant and visually appealing apps, and access to a modern language with strong community support.
* **Android Studio**
* Android Studio offers a comprehensive and user-friendly environment for building Android applications. Its features streamline the development process, from writing code and testing to debugging and deployment.

* + 1. **Sign-Up Screen**

****

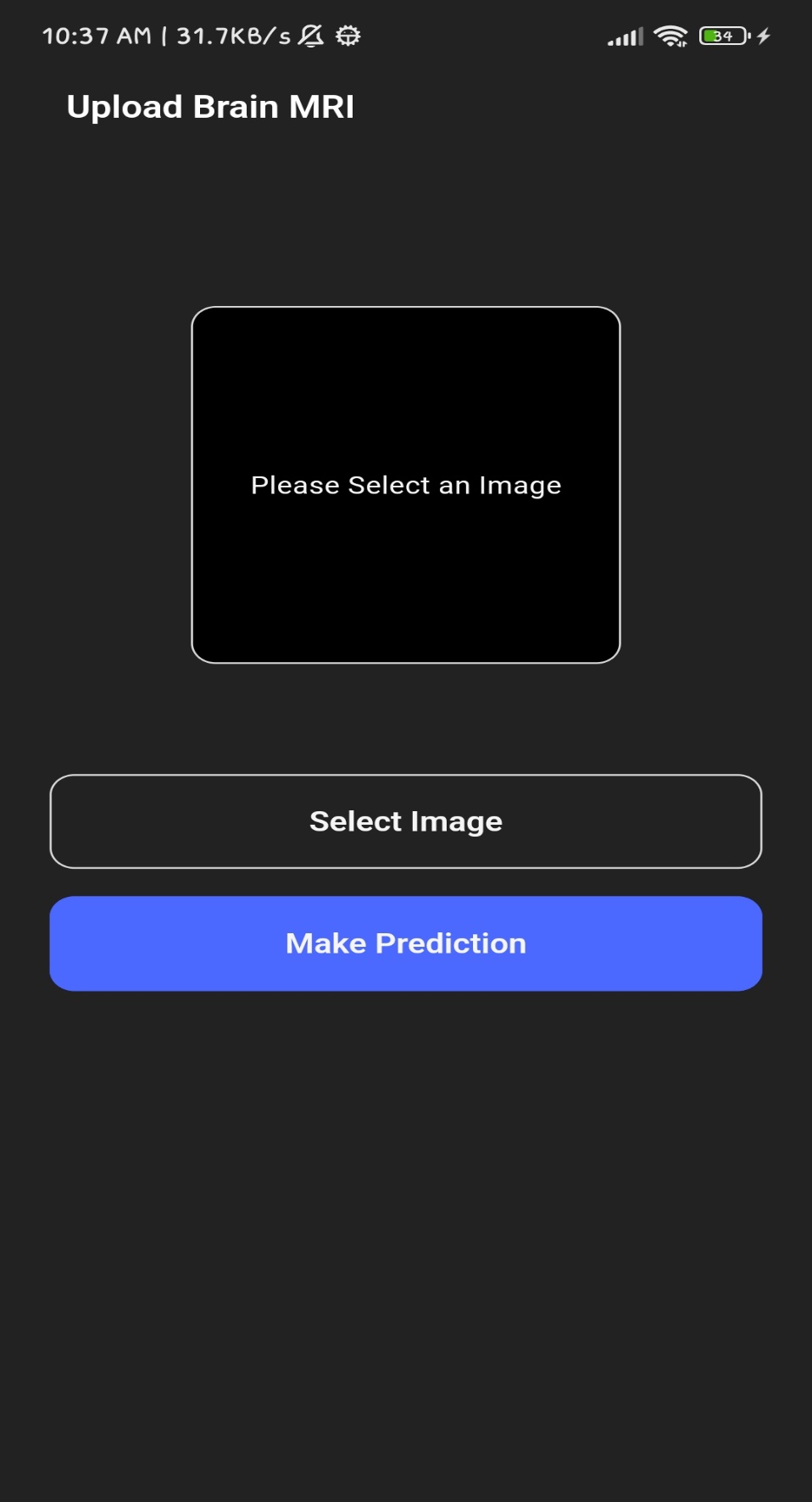
[Fig. 7.1.1 – Sign-up Screen]

* + 1. **Sign-in Screen**



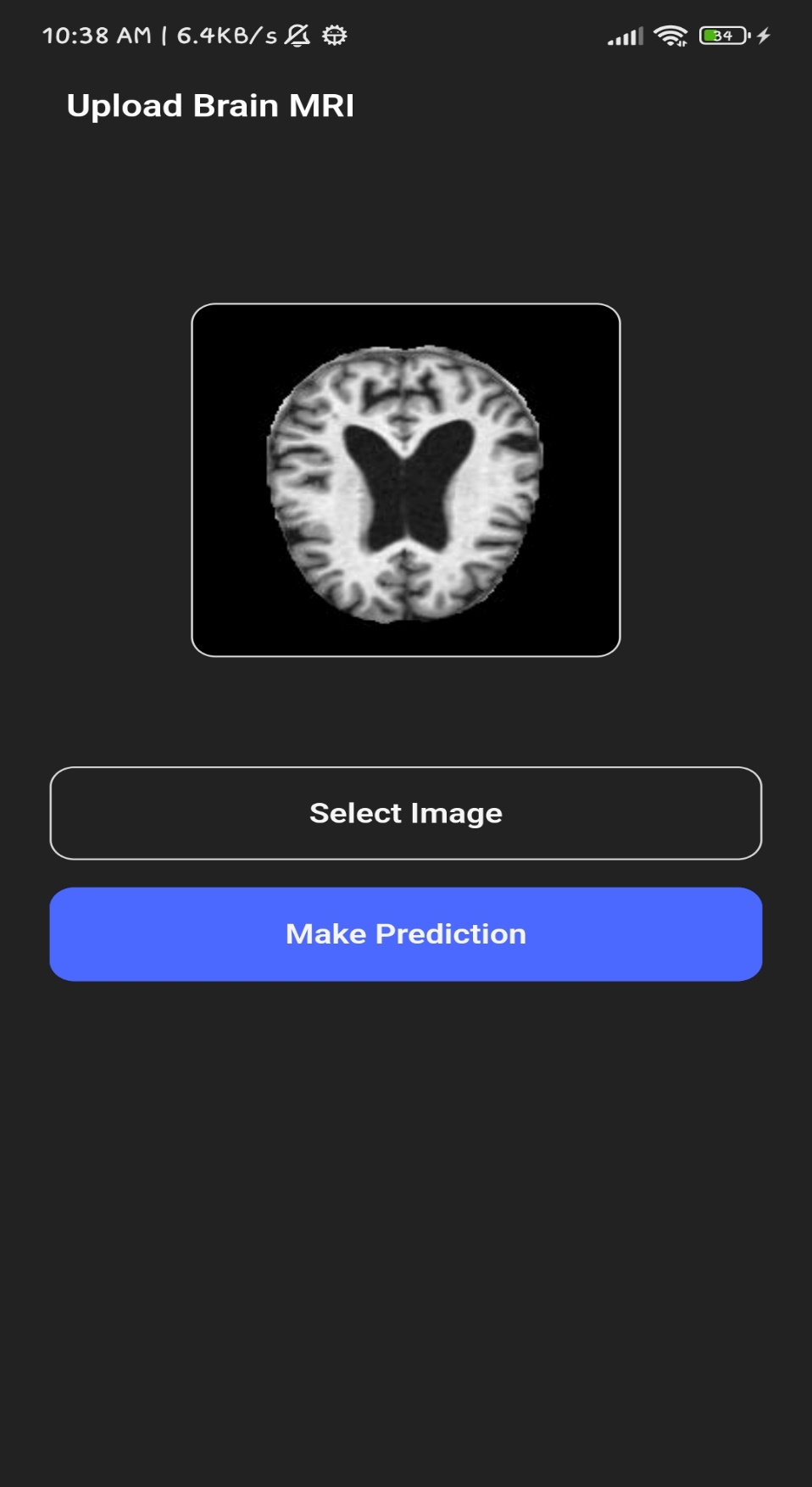
[Fig. 7.1.2 – Sign-in Screen]

* + 1. **Screen for Upload an Image**

****

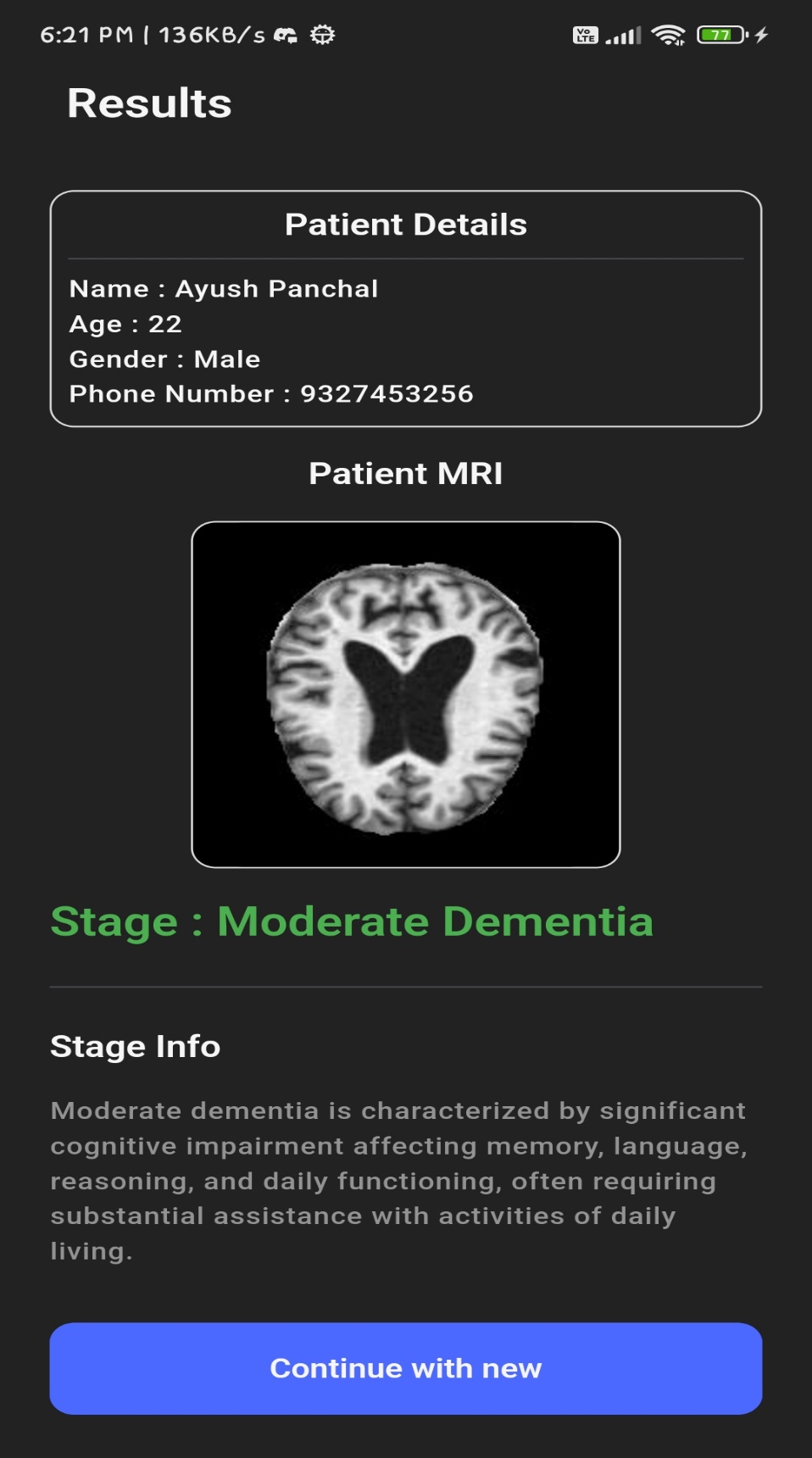
[Fig. 7.1.3 – Screen for Upload an Image]

* + 1. **Screen of Uploaded an Image**

****

[Fig. 7.1.4 – Screen of Uploaded Image]

* + 1. **Result Screen**

****

[Fig. 7.1.5 – Result Screen]

* 1. **BACKEND IMPLEMENTATION**
* To Implement Backend of the application, various techniques are used including:
* **Image Processing**

In the context of the Alzheimer disease detection project, image processing plays a pivotal role. It involves a series of techniques applied to raw MRI scans to enhance their quality and prepare them for analysis. Techniques like resizing, normalization, and data augmentation are employed to ensure that images are in a consistent format and that the model can effectively learn from them. Image processing helps improve the accuracy and robustness of the disease detection system by standardizing and diversifying the training data.

* **Transfer Learning**

Transfer learning is a critical component of the project's success. It allows the project to leverage pre-trained Convolutional Neural Networks (CNNs) that have already learned essential features from extensive datasets. By fine-tuning these pre-trained models on Alzheimer disease images, the project can expedite model training and achieve higher accuracy. Transfer learning helps bridge the gap between generic image recognition and Alzheimer disease detection, making the system more efficient and effective in identifying diseases across various Alzheimer species.

* **Convolutional Neural Networks**

Convolutional Neural Networks, or CNNs, serve as the project's backbone for image analysis. These specialized deep learning models are designed to automatically extract relevant features from images. In the Alzheimer disease detection system, CNNs play a crucial role in recognizing patterns and textures associated with various diseases. By breaking down complex images into meaningful components, CNNs enable the model to make accurate disease classifications. Their ability to learn hierarchical features from data makes CNNs well-suited for image-based tasks like Alzheimer disease detection, where visual patterns are indicative of brain health.

* In this project, various modules and packages of Python language are used:
* **TensorFlow**

TensorFlow, an open-source library by Google, is a powerful tool for machine learning tasks. It allows you to build and train various models, from simple to complex, using different levels of abstraction. Once you have created your model, TensorFlow offers ways to deploy it on diverse platforms, making it versatile for real-world applications. Additionally, TensorFlow’s ability to handle large datasets and leverage powerful hardware makes it suitable for complex machine learning projects.

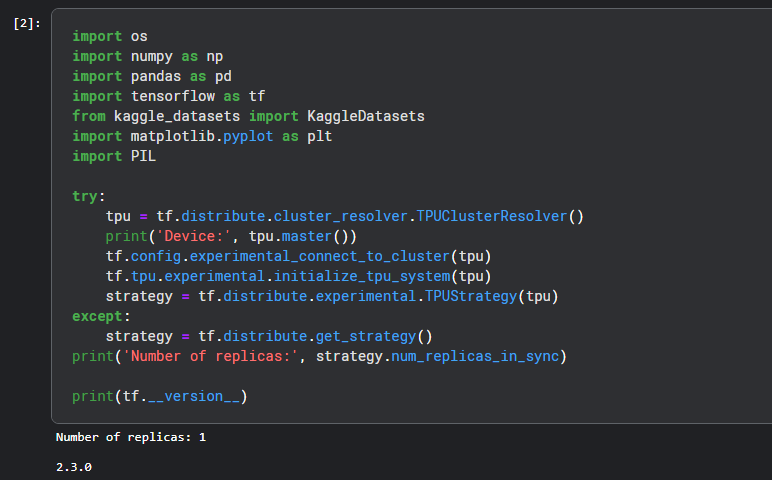
* **Scikit-Learn**

Scikit-learn, a free and user-friendly Python library, offers a treasure trove of tools for machine learning. Beginners and experts alike appreciate its simplicity, allowing them

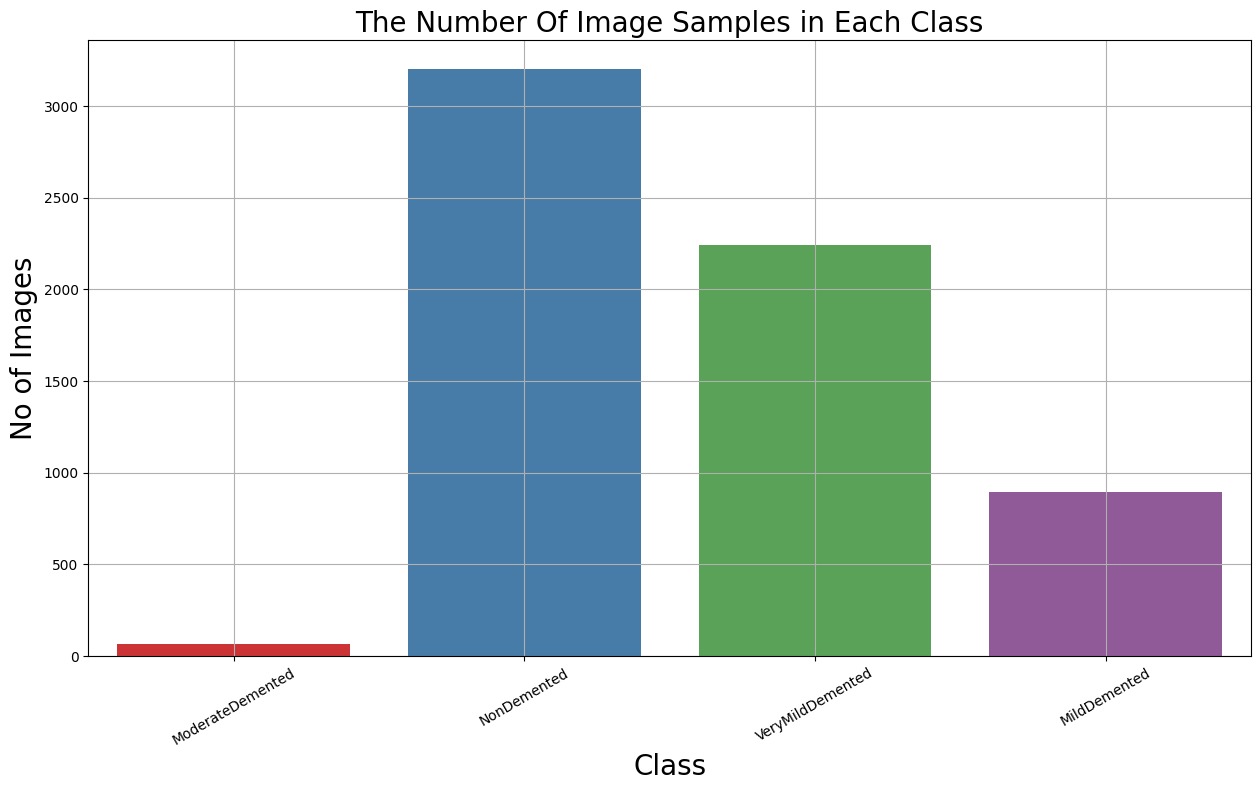
to implement various algorithms efficiently, without writing everything from scratch. This focus on ease of use empowers you to solve problems quickly by choosing the right tool from scikit-learn’ s broad toolkit, which encompasses algorithms for classification, regression, and more. Built on top of powerful Python libraries, scikit-learn integrates seamlessly with the data science landscape, making it a favourite for machine learning projects.

* **Flask-API**

Flask API lets you build web APIs using Python's Flask framework. Imagine it as a way for apps to talk to each other. Flask, known for its simplicity, provides the core tools to handle these conversations. You can create APIs that respond to requests from other applications, like sending or receiving data in formats like JSON. This makes Flask a great choice for building lightweight and flexible web services.

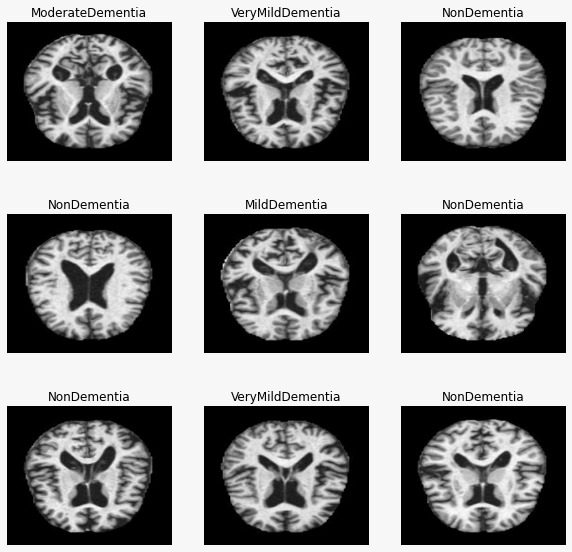
* + 1. **Importing necessary dependencies**

[Fig. 7.2.1 – Importing necessary dependencies]

* + 1. **Number of Images per Class :**

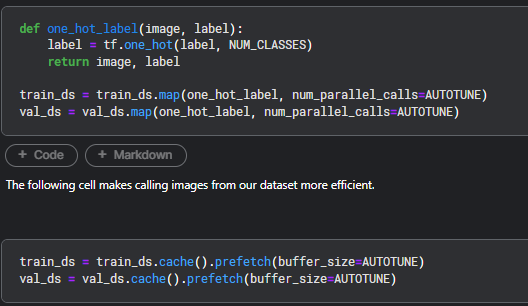
[Fig. 7.2.2 – Number of Images per class]

* + 1. **Visualization of MRI’s:**

****

[Fig. 7.2.3 – Visualization of MRI’s]

* + 1. **Feature Engineering one hot encoding:**

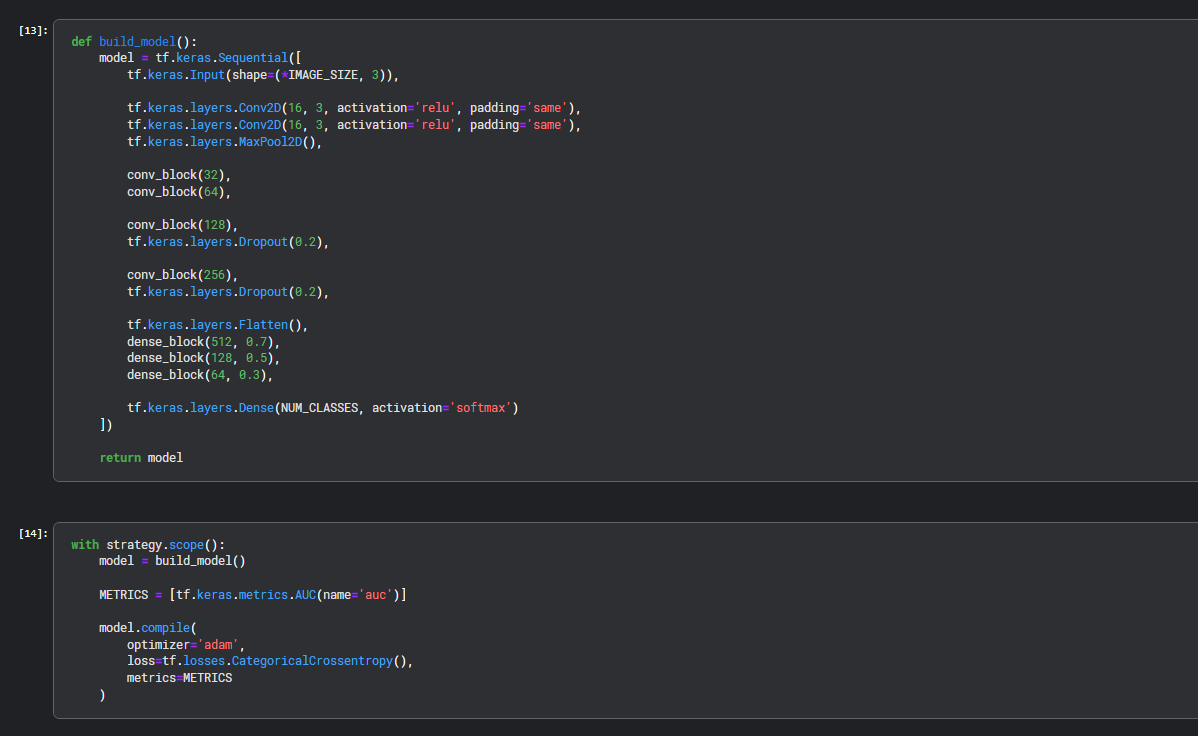


[Fig. 7.2.4 – Feature Engineering One hot encoding]

* + 1. **Building ML Model\_Phase-1:**

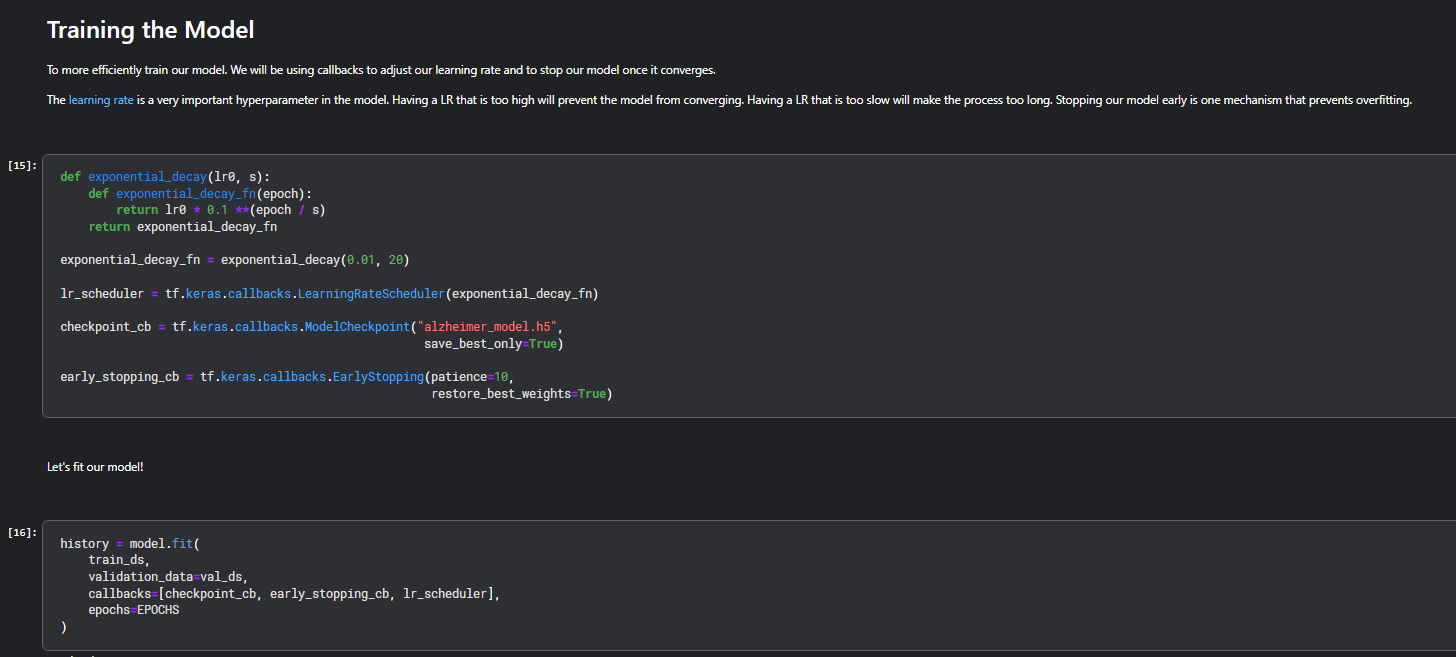
****

[Fig. 7.2.5 – Building ML Model\_Phase-1]

* + 1. **Building ML Model\_Phase-2:**

[Fig. 7.2.6 – Building ML Model\_Phase-2]

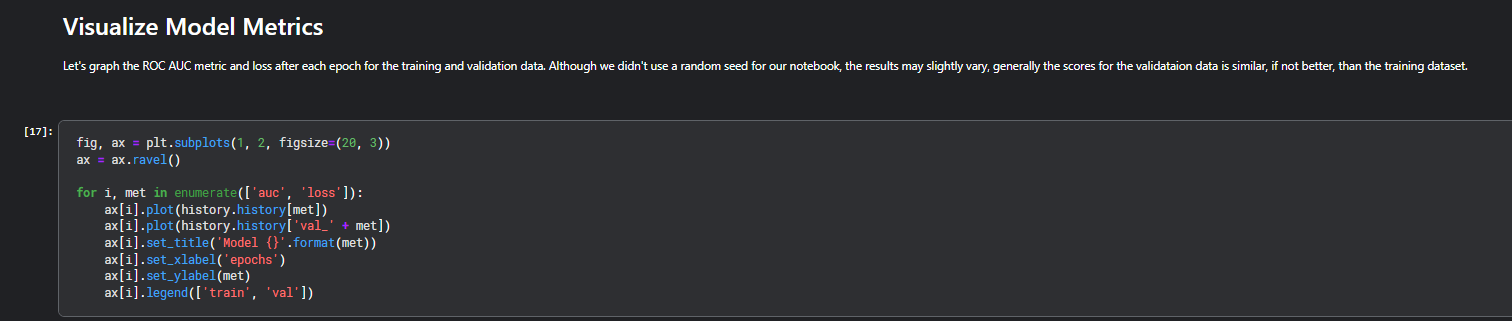
* + 1. **Model Training:**

****

[Fig. 7.2.7 – Model Training]

* + 1. **Visualizing model performance with respect to epochs:**

In this phase, each graphs show model’s performance over the number of epochs.

****

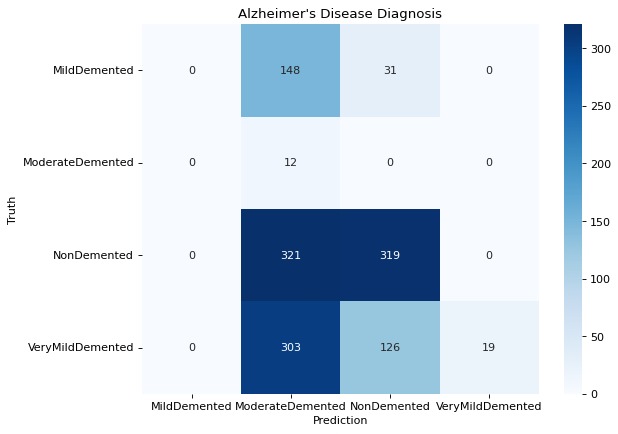
[Fig. 7.2.8 – Visualizing model performance with respect to training Phase]

* + 1. **Model Evaluation:**

****

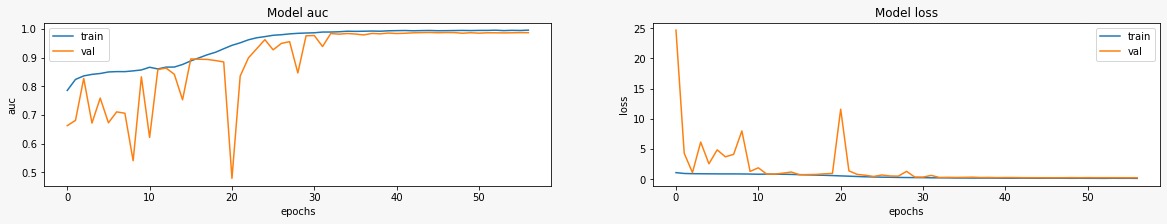
[Fig. 7.2.9 – Model Evaluation]

* + 1. **Confusion Matrix:**

**

[Fig. 7.2.10 – Confusion Matrix]

* + 1. **Testing graph:**



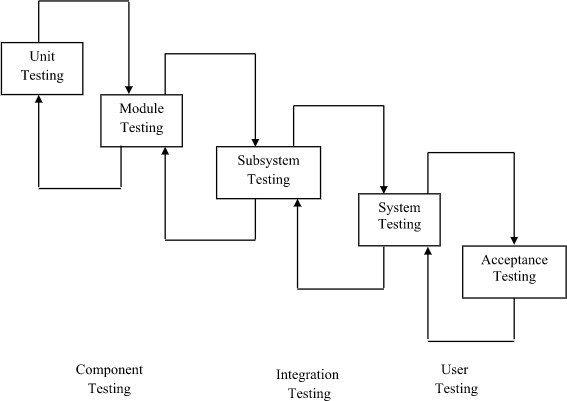
[Fig. 7.2.11 – Testing graph]

**CHAPTER 8**

**SYSTEM TESTING**

1. **SYSTEM TESTING**
   1. **Test plan:**

* Testing has a dual function; it is used to establish the presence of defects in a program and it is used to judge whether or not the program is usable in practice. Thus, testing is useful for validation and verification, which ensure that software conforms to its specification and meets the needs of the software customer.
* I restored Alpha testing, usually comes in after the basic design of the program has been completed. The project scientist will look over the program and make suggestion or given ideas to us to improve or to correct the design. They also report and give ideas to help get rid of or work around any MJOR problems. There is bound to be a number of bugs after program is created and they are most likely to get known to the developers the alpha testing. I carried out testing process in four stages i.e.
* Unit testing
* Module testing
* Subsystem testing
* System testing
* In another method called Black Box or functional testing. I am concerned about the output of the module and software, i.e. whether the application gives proper output as per requirement or not. In another words, these testing aims to test a given program’s behaviour.
* Against its specification without making any reference to the internal structure of the program or the algorithm used. Therefore, the source code id not needed, and so even purchased modules can be tested. The program just gets a certain input, and its functionality is examined by observing the output. This can be done in following way:
* Input
* Interface
* Processing
* Output
* The tested program gets a certain input or the input is observed. Then the product does its job & generates a certain output, which is collected by a second interface. This result is then compared to the expected output, which has been determined before the test.
* While white Box testing was used as an important primary testing approach; code is inspected to see what it does, tests are designed to exercise the code. Code is tested using code: scripts, drivers, stubs, etc. are employed to directly interface with and drive the code.
* Testing & debugging is done in two steps in our project. Actually, the testing process started right from the word go in the project life spans each and every module was being worked upon.



[Fig. 8.1 – Flow of Testing]

* 1. **Test Results and Analysis:**

An Alzheimer's prediction application might achieve high accuracy, like 82%, in identifying healthy brain patterns versus those with Alzheimer's. However, it might be better at identifying healthy brains (high precision) and miss some early cases of Alzheimer's (lower recall). This could be because the app prioritizes avoiding false positives. The application might need adjustments to better balance identifying early Alzheimer's cases while maintaining good overall accuracy. Despite this, if the app performs well on unseen data and is user-friendly, it has promise. Further improvements could focus on finding a better balance in disease classification and making results come out faster.

**CHAPTER 9**

**LIMITATIONS AND**

**FUTURE**

**ENHANCEMENT**

1. **LIMITATIONS AND FUTUTRE ENHANCEMENT**
   1. **Limitations**

Despite the scope, features and modules aforementioned, there are a few limitations in

**“MindGuard: AI – Unveiling Alzheimer”,** these are listed below:

* Data Dependence
* Image Quality Matters
* Psychological Impact of Predictions
* Social Stigma and Discrimination
* Limited Offline Functionality
  1. **Summary of Project Work:**

**Goal:** This project aims to develop a mobile application that leverages machine learning to predict the risk of Alzheimer's disease.

* **Benefits:**
* Early detection allows for earlier intervention and treatment planning.
* Personalized risk assessments inform preventative measures and lifestyle choices.
* Empowers individuals to take a proactive approach to their brain health.
* **Technical Approach:**
* **Image capture:** Users scans MRI of brain.
* **Machine learning model:** CNN or similar models to analyse brain MRIs and predict risk of Alzheimer’s and potential use of deep learning for complex data integration.
* **Disease prediction:** The app suggests the risk of Alzheimer's disease
* **Limitations:**
* **Data Accuracy:** Relies on the accuracy of user-provided data and quality of training data for machine learning models.
* **Early Detection stage:** Predicts risk, not a definitive diagnosis. Confirmation through medical procedure might be needed.
* **Algorithmic Bias:** Machine learning models can inherit biases from training data, potentially impacting prediction accuracy for certain demographics.
  1. **Future Enhancement:**
* **Deeper Data Dive:** Go beyond cognitive tests. Integrate genetics, bloodwork, lifestyle factors, and even voice analysis to create a richer risk profile.
* **Smarter Machine Learning:** Utilize deep learning for complex data patterns and multimodal learning to combine different data types for a more robust prediction.
* **Personalized Approach:** Move from general risk scores to tailored assessments based on individual factors like age, family history, and habits. Predict not just onset, but also disease progression to guide future care.
* **Wearable Integration:** Continuously monitor cognitive function, sleep, and activity through wearables and sensors for early detection and disease tracking.
* **User-Friendly Design:** Provide clear, actionable information about risk and predictions, while prioritizing user privacy and data security.

**CHAPTER 10**

**REFERENCE**

1. **REFERENCES**

Links Following:

1. National Library of medicine: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10093003/>
2. <https://www.kaggle.com/datasets/tourist55/alzheimers-dataset-4-class-of-images>
3. Tensor flow documentation: <https://www.tensorflow.org/learn>
4. Flutter Documentation: <https://docs.flutter.dev/\>
5. Firebase Documentation: <https://firebase.google.com/docs>
6. Flask Documentation: <https://flask.palletsprojects.com/en/3.0.x/>
7. Kaggle platform: <https://www.kaggle.com/>
8. Udemy Course : <https://www.udemy.com/home/my-courses/learning/>