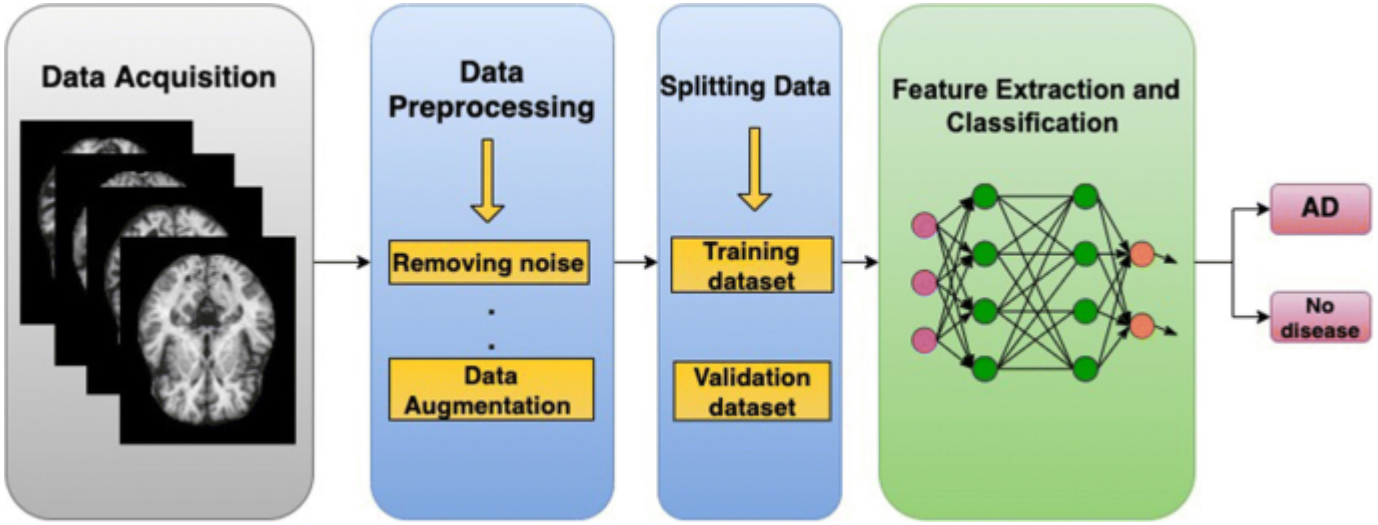
Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	06 May 2023
Team ID	NM2023TMID00113
Project Name	Cognitive Care: Early Intervention for Alzheimer's Disease

Technical Architecture:



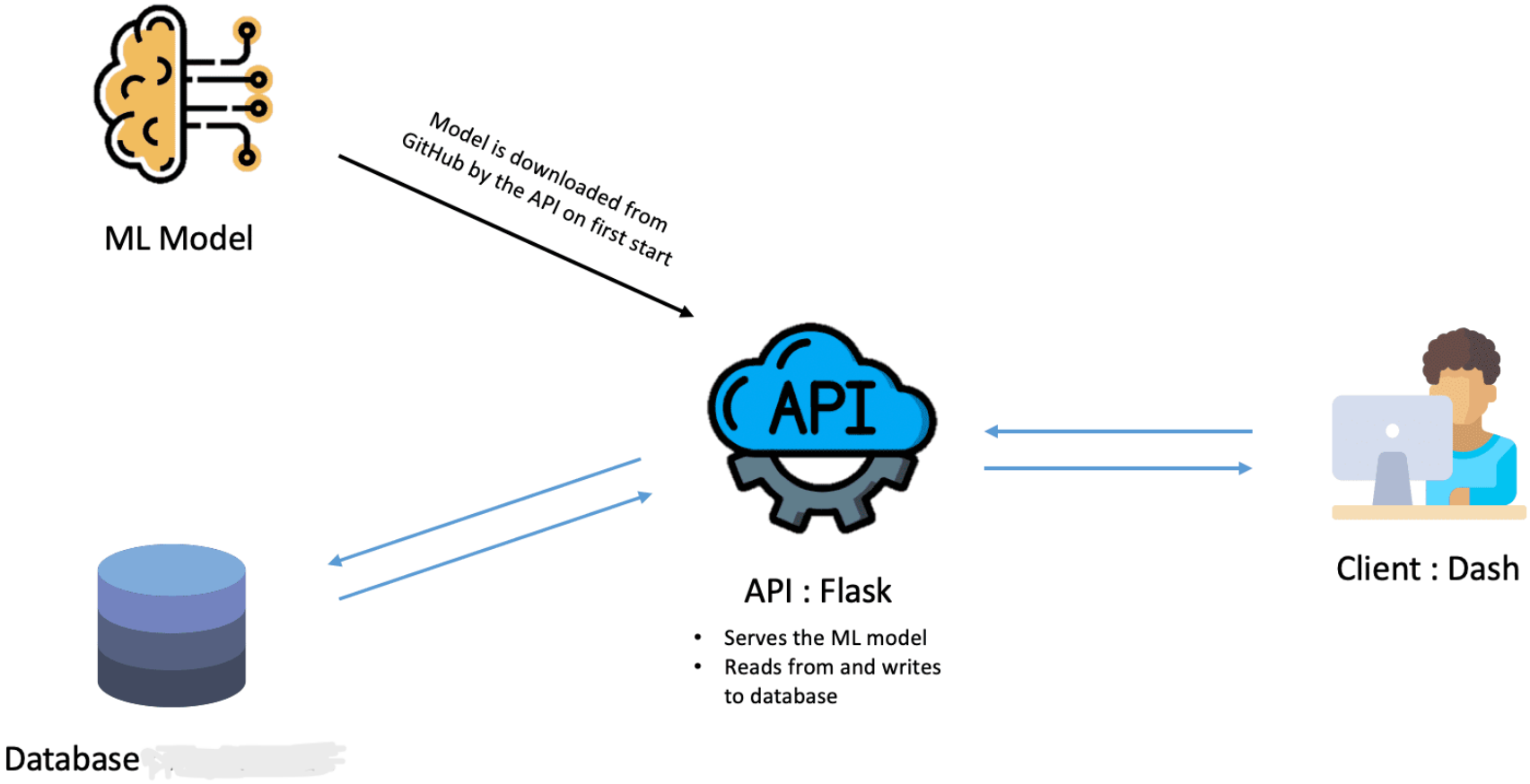

ML Model
Model is downloaded from GitHub by the API on first start
API
API : Flask
• Serves the ML model
• Reads from and writes to database
Client : Dash
Database

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The User interacts through a classic web interface	HTML, CSS, JavaScript, Python, Jinja2 Templating, Bootstrap Styling
2.	Website Backend	Backend URL/View APIs	Python, Flask
3.	Alzheimer Classification Model	To Train and run a deep learning model to classify MRI images	Tensorflow, Keras API, Xception
4.	To Preprocess MRI Image	Preprocessing the image before feeding to inference model	Pillow, OpenCV, ImageDataGen
5.	Database	To store user data if necessary (optional from customer)	Sqlite DB
6.	Cloud Database	Database Service on Cloud	IBM Cloud Object Storage
7.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
8.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
9.	Infrastructure (Server / Cloud)	The Deep Learning Model file is hosted in a deployment space in the cloud to access anywhere using the Watson Machine Learning Client	IBM Watson Studio Machine Learning

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep Learning Libraries based on Numpy and a Web Server Gateway Interface for Python as a micro-framework	Flask, Tensorflow, OpenCV, Pillow, Bootstrap
2.	Security Implementations	Solution is to be hosted over the web or on customer server premises. Typical web safety protocols apply	CSRF protections will suffice.
3.	Scalable Architecture	It is made with a Flask micro-framework which may be extended with any number of compatible and extensible libraries such as Flask-Login for authentication, Flask-Restful for REST API development, Flask-SQLAlchemy for ORM's etc.	Flask
4.	Availability	Multiple strategically placed servers with load balancers could be used to make the service available at any time. Our problem statement is not a immediate life-or-death situation	Server-side software for efficient server-side resource allocation that is optional during deployment. IBM Watson Studio Deployment Asset Management services are ideal for this functionality.
5.	Performance	Alzheimer's is a rare disease and number of users is not as such as a regular application	N/A