**RATHINAM TECHNICAL CAMPUS**

**COGNITIVE CARE: EARLY INTERVENTION FOR**

**ALZHEIMER’S DISEASE**

**Team ID :** NM2023TMID11242

**Team Size :** 4

**Team Leader :** HARIKUMAR S

**Team member :** ABIRAMI J

**Team member :** NIVETHA S

**Team member :** DEEPIKA SRI K

**REPORT**

|  |  |
| --- | --- |
| **Team ID** | NM2023TMID11242 |
| **Project Name** | Cognitive Care: Early Intervention for Alzheimer's Disease |

**Project Report**

**1.** **INTRODUCTION**

1.1 Project Overview

1.2 Purpose

**2.** **IDEATION & PROPOSED SOLUTION**

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

**3.** **REQUIREMENT ANALYSIS**

3.1 Functional requirement

3.2 Non-Functional requirements

**4. PROJECT DESIGN**

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

**5. CODING & SOLUTIONING (Explain the features added in the project along with code)**

5.1 Feature 1

5.2 Feature 2

5.3 Database Schema (if Applicable)

**6. RESULTS**

6.1 Performance Metrics

**7. ADVANTAGES & DISADVANTAGES**

**8. CONCLUSION**

**9. FUTURE SCOPE**

**10. APPENDIX**

Source Code

**1.INTRODUCTION**

**1. Project Overview:**

The project aims to enhance understanding and address the challenges associated with Alzheimer’s disease. It encompasses research, education, awareness campaigns, and support initiatives to improves the lives of individuals with Alzheimer’s, their families, and caregivers.

**2. Objectives:**

* Early Detection and Diagnosis: Increase the rate of early detection and accurate diagnosis of Alzheimer's disease through improved screening methods, awareness campaigns, and healthcare provider education.
* Timely Interventions: Ensure timely initiation of interventions, including medication, cognitive stimulation, lifestyle modifications, and support services, upon diagnosis to slow down the progression of the disease and improve cognitive function.
* Education and Support: Provide comprehensive education and support to individuals with Alzheimer's, their families, and caregivers to enhance their understanding of the disease, improve coping strategies, and promote overall well-being.
* Accessible Healthcare: Improve access to specialized healthcare professionals, diagnostic tools, and treatment options for individuals with Alzheimer's disease, reducing wait times and ensuring equitable healthcare services.
* Caregiver Support: Develop and implement caregiver support programs, including education, respite care, and mental health services, to address the unique challenges faced by caregivers and enhance their ability to provide quality care.
* Research and Innovation: Promote and support research initiatives focused on Alzheimer's disease, including clinical trials, to advance knowledge, develop new treatment modalities, and contribute to scientific advancements in the field.
* Quality of Life: Enhance the overall quality of life for individuals with Alzheimer's by addressing their cognitive, emotional, and social needs, and fostering a supportive and inclusive environment.

**3. Key Activities:**

**Awareness Campaigns:**

Develop and implement public awareness campaigns to educate the general population about Alzheimer's disease, its early signs, and the importance of early intervention. Conduct community outreach programs, workshops, and informational sessions to raise awareness among specific target groups, such as healthcare professionals, caregivers, and older adults.

**Caregiver Support and Education:**

Develop caregiver support programs that provide information, resources, and training to help caregivers understand the disease, manage symptoms, and cope with the challenges of caregiving. Organize support groups, counseling services, and respite care options to address the emotional and practical needs of caregivers.

**Research and Clinical Trials:**

Facilitate and support research studies and clinical trials focused on early interventions for Alzheimer's disease. Promote collaboration between research institutions, pharmaceutical companies, and healthcare providers to advance scientific understanding and develop innovative treatments.

**4. Expected Outcomes:**

- Increased rates of early detection and accurate diagnosis of Alzheimer's disease, allowing for timely interventions and support.

- Enhanced understanding of Alzheimer's disease among patients, caregivers, and healthcare providers, leading to better-informed decision-making and proactive management of symptoms.

- Enhanced accessibility to specialized healthcare services, reducing wait times for diagnosis and interventions, and ensuring equitable care for individuals with Alzheimer's disease.

- Improved overall quality of life for individuals with Alzheimer's and their caregivers through comprehensive support, symptom management, and community engagement.

**5. Timeline and Resources:**

Develop a comprehensive timeline for the project, including milestones, deliverables and allocation of resources such as funding, personnel and partnerships. Continuously monitor and evaluate progress to ensure project success and make adjustment as necessary.

**1.2. Purpose**

The purpose of studying and addressing Alzheimer’s disease is multi-faceted and encompasses several important objectives:

**Understanding the Disease:** One of the primary purposes is to gain a deeper understanding of Alzheimer's disease, including its causes, progression, and underlying mechanisms. By studying the disease, researchers aim to unravel the complex processes involved in Alzheimer's and identify potential targets for intervention and treatment.

**Early Detection and Diagnosis:** Studying Alzheimer's disease enables the development of better diagnostic tools and techniques for early detection. Early diagnosis is crucial as it allows for timely interventions, access to appropriate care, and the potential to slow down the progression of the disease.

**Treatment and Management:** Research on Alzheimer's disease aims to discover effective treatment options to slow the progression of cognitive decline, improve quality of life, and alleviate symptoms. This includes exploring pharmacological interventions, lifestyle modifications, cognitive interventions, and support services for individuals with Alzheimer's and their caregivers.

**Caregiver Support and Education:** Studying Alzheimer's disease also involves understanding the impact of the disease on caregivers and developing support systems to assist them in managing the challenges associated with caregiving. This includes providing educational resources, respite care options, and mental health support to caregivers.

**Public Health Planning:** Research on Alzheimer's disease contributes to public health planning and policy development. It helps policymakers and healthcare systems understand the prevalence, impact, and future projections of Alzheimer's, enabling them to allocate resources, develop appropriate healthcare infrastructure, and implement preventive measures at a population level.

**2. IDEATION & PROPOSED SOLUTION**

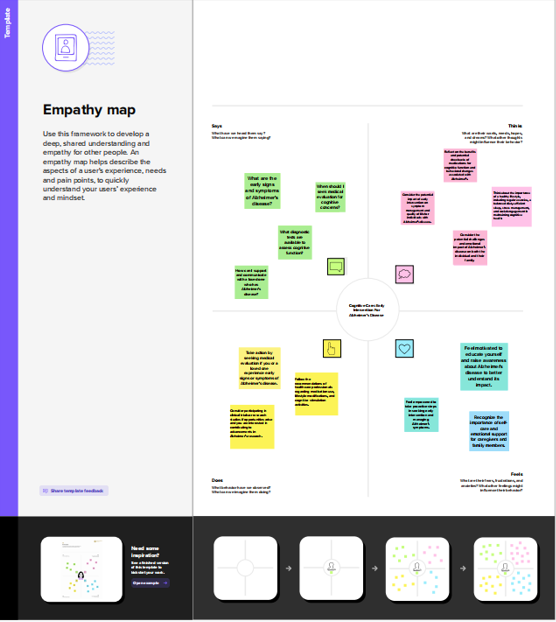
**2.1 Problem Statement Definition:**

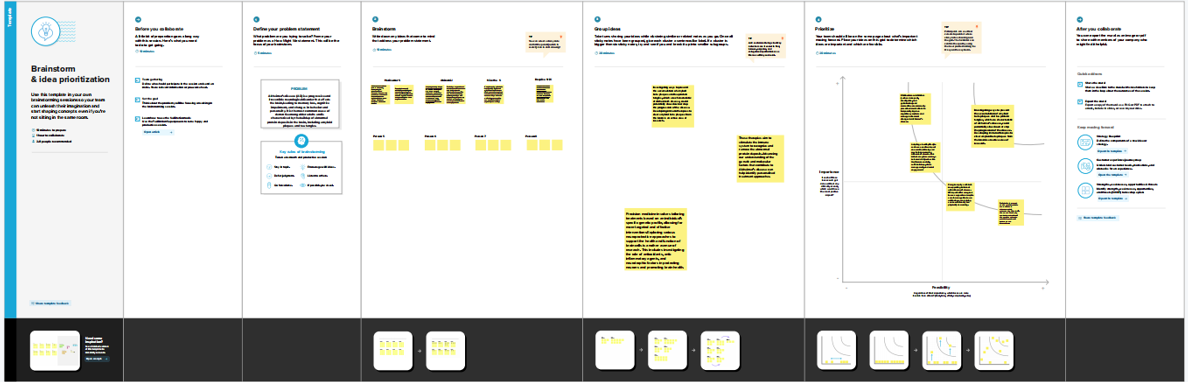
Alzheimer's disease (AD) is a progressive and irreversible neurological disorder that affects the brain, leading to memory loss, cognitive impairment, and changes in behavior and personality. It is the most common cause of dementia among older adults and is characterized by the buildup of abnormal protein deposits in the brain, including amyloid plaques and tau tangles.

The exact cause of Alzheimer's disease is not yet fully understood, but it is believed to be influenced by a combination of genetic, environmental, and lifestyle factors. Age is also a significant risk factor, with the risk of developing the disease increasing significantly after the age of 65.The early symptoms of Alzheimer's disease may include mild memory loss, difficulty with problem-solving, and changes in mood or behavior. As the disease progresses, these symptoms become more severe, with individuals experiencing significant memory loss, difficulty communicating, and a loss of the ability to perform daily activities.

By using deep learning models like Xception to analyze medical imaging data, it may be able to identify early signs of Alzheimer's disease before symptoms become severe. This can help healthcare providers to provide early treatment and support for patients and their families, ultimately leading to better outcomes for all involved.

**2.2. Empathy Map:**

** 2.3. Ideation and Brainstorming:**

****

**2.4. Proposed Solution:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | The problem is the inadequate provision of cognitive care and early intervention for Alzheimer's disease, resulting in delayed diagnosis, limited access to specialized healthcare, and a lack of awareness among individuals and families. |
|  | Idea / Solution description | Provide resources and support services for caregivers, including education, respite care, and mental health support, to help them navigate the challenges of caring for individuals with Alzheimer's disease. |
|  | Novelty / Uniqueness | The uniqueness of this solution lies in its comprehensive approach, combining awareness, accessibility, and early interventions, which addresses the multifaceted challenges of cognitive care and early intervention for Alzheimer's disease. |
|  | Social Impact / Customer Satisfaction | Increased awareness and education about Alzheimer's disease lead to early detection, enabling individuals to seek appropriate care and support, reducing the burden on families and healthcare systems. Individuals and families affected by Alzheimer's disease benefit from the support provided through public campaigns, accessible healthcare, and early interventions, resulting in improved satisfaction with the healthcare system. |

**3.** **REQUIREMENT ANALYSIS**

**3.1. Functional Requirement**

**1. User Registration:**

* Registration through form
* Registration through Gmail.
* Registration through LinkedIN.

**2. User Confirmation:**

* Confirmation via Email.
* Conformation via OTP

**3. Data Collection:**

The system should be able to collect relevant data from individuals, such as demographics, medical history, cognitive assessments, genetic information, and lifestyle factors.

**4.Preprocessing and Feature Extraction:**

The system should preprocess the collected data, clean and normalize it and extract relevant features for Alzheimer’s disease prediction, such as cognitive test scores, biomarkers, or genetic markers.

**5.Model Training and Validation:**

The system should be able to train the predictive models using labelled data, such as historical patient data, and validate the models to assess their performance and accuracy.

**3.2 Non-Functional requirements**

**1.Usability:**

* The system should have an intuitive and user-friendly interface for healthcare professionals and caregivers.
* The system should provide clear and understandable instructions or guidance.
* The system should support accessibility features to accommodate users with disabilities.

**2.Security:**

* The system should adhere to privacy regulations and protect sensitive patent information.
* The system should implement appropriate access controls and authentication mechanism.

**3. Reliability:**

* The system have high availability to ensure continuous access for healthcare professionals and caregivers.
* The system should minimize downtime and be recover quickly in the event of failures or disruptions.

**4.Performance:**

* The system should provide real time or near real time response for critical operations.
* The system should handle a large volume of data efficiently.
* The system should have low latency when processing and retrieving information.

**5.Availability:**

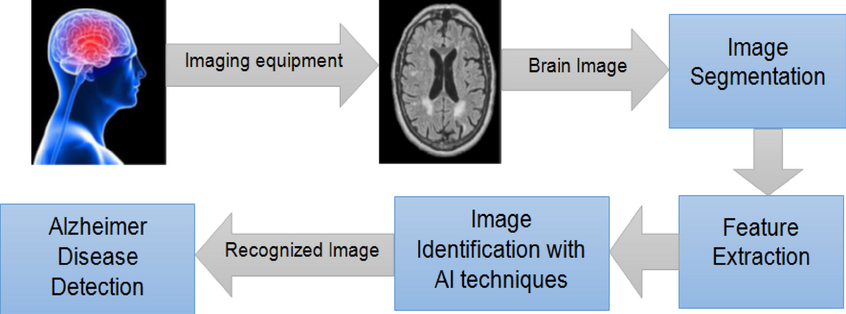
* The System should be compatible with various devices, operating systems and wed browsers.
* The system should integrate seamlessly with other healthcare systems or electronic health record(HER) systems.

**4. PROJECT DESIGN**

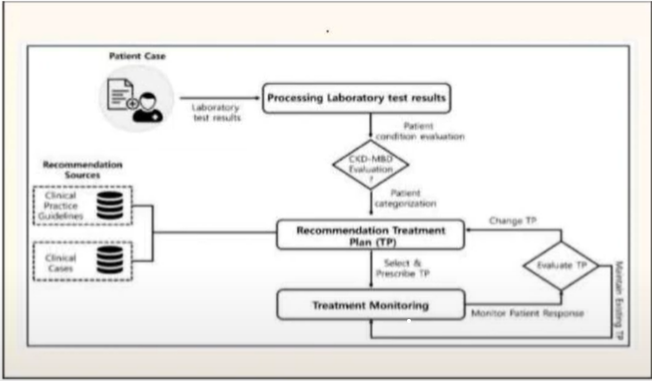
**4.1 Data Flow Diagrams**

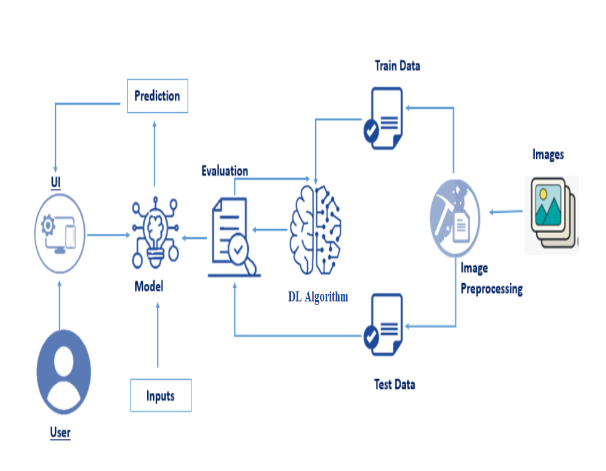
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

**Flow Diagram:**



**Data flow Diagram:**



**4.2. Solution & Technical Architecture:**

**4.3 User Stories**

1. As a caregiver, I want access to educational resources and support groups so that I can better understand and cope with the challenges of caring for my loved one with Alzheimer’s disease.

2. As a person with early stage Alzheimer’s, I want a mobile application that provides reminders for daily tasks, medication schedules, and important appointments to help me maintain my independence and stay organized.

3. As a family member, I want an online platform where I can securely store and share my loved one’s medical information, including diagnostic, medications, and contact details of healthcare professionals to ensure coordinates care among the care giving team.

4. As a researcher, I need a comprehensive database of anonymized patient data and biomarkers to facilitate my study on identifying early diagnostic markers and potential treatment targets for Alzheimer’s disease.

5. As a healthcare professional, I want access to interactive training modules that provide evidence-based guidelines and best practices for diagnosing, managing and providing person- centered care for individuals with Alzheimer’s disease.

6. As a person with Alzheimer’s disease, I want a safe and stimulating environment at my care facility, with access to memory- enhancing activities, social engagement opportunities, and a compassionate staff trained in dementia care.

**5. CODING & SOLUTIONING (Explain the features added in the project along with code)**

import numpy as np

import os

from keras.preprocessing import image

import pandas as pd

import ev2

import tensorflow as tf

#Flask utils

from flask import Flask, request, render-template from werkzeug.utils import secure\_filename

from tensorflow.python.keras.backend import set-session from tensorflow.python:keras.models import load\_model

global graph

tf.disable\_v2\_behavior() graph=tf.get\_default\_graph()

app=Flask(\_\_\_name\_set\_session(sess)model=load\_model(‘adp.h5’)  
#Load your trained model

@app.route(‘/’,methods=[‘GET’])def index():

#Main page

return render\_template(‘alzheimers.html’)

@app.route(‘/predict1’,methods=[‘GET’])

def predict1():

#Main page

return render\_template(‘alzpre.html’)

@app.route(‘/predict’,methods=[‘GET’,’POST’]) def upload():

if request.method==’POST’:#Get the file from post request

f=request.files[‘images’]

#Save the file to./uploads basepath=os.path.dirname(\_\_file\_\_)

file\_path=os.path.join(basepath, ‘upload’, secure\_filename(f.filename)) img =image.load\_img(file\_path,target\_size=(224,224))

f.save(file\_path)

x=image.img\_to\_array(img)

**6.RESULTS**

**6.1 Performance Metrics**

Performance metrics for Alzheimer’s disease initiatives can help assess the effectiveness, efficiency and impact of interventions, research studies and support services.

1.Diagnosis Rate:

- Percentage of individuals with suspected cognitive impairment who receive a timely and accurate diagnosis of Alzheimer’s disease.

- Number of individuals screened for cognitive impairment and subsequent diagnosis rates.

2. Timely Intervention:

- Percentage of individuals diagnosed with Alzheimer’s disease who receive appropriate medical, cognitive, and psychosocial interventions within a specified timeframe.

3. Caregiver Satisfaction:

- Surveys or feedback from caregivers to assess their satisfaction with the provided support services, resources and education.

- Measurement of caregiver burden and stress levels before and after accessing support programs.

**7. ADVANTAGES & DISADVANTAGES**

Advantages:

Slowing Disease Progression: Early intervention may help slow down the progression of Alzheimer's disease. Initiating treatment and interventions early can potentially delay the onset of more severe symptoms, allowing individuals to maintain a higher level of cognitive function and independence for a longer period.

Improved Quality of Life: Early intervention can contribute to improving the overall quality of life for individuals with Alzheimer's disease. It can help manage symptoms, reduce behavioral and psychological symptoms, and enhance daily functioning. This may lead to better emotional well-being, improved social interactions, and enhanced engagement in meaningful activities.

Better Response to Treatment: Starting treatment early may lead to better response rates to medications and interventions. At the early stages of the disease, individuals may have a higher likelihood of benefiting from available pharmacological treatments, behavioral interventions, and cognitive rehabilitation techniques.

Caregiver Support and Education: Early intervention provides an opportunity for caregivers to receive support and education. Caregivers can learn about the disease, effective caregiving strategies, and available resources, helping them navigate the challenges associated with Alzheimer's disease. Early involvement can also facilitate the development of coping mechanisms and the establishment of support networks.

Disadvantages:

Financial Considerations: Early intervention for Alzheimer's disease may involve medical consultations, diagnostic tests, and potentially expensive treatments. Access to healthcare and financial considerations can be barriers to receiving timely and appropriate interventions, particularly for individuals without adequate insurance coverage or financial resources.

Emotional Impact: Receiving an early diagnosis of Alzheimer's disease can be emotionally distressing for individuals and their families. It may lead to feelings of fear, anxiety, and sadness about the future. Coping with the diagnosis and adapting to lifestyle changes may be particularly challenging at the early stages when individuals are still adjusting to the reality of the disease.

**8. CONCLUSION**

In conclusion, early intervention for Alzheimer's disease offers several potential advantages and disadvantages. On the positive side, early intervention may help slow the progression of the disease, improve quality of life, increase treatment response rates, and provide support and education for caregivers. However, there are challenges associated with early diagnosis, limited treatment options, emotional impact, and financial considerations. Despite these challenges, early intervention remains an important area of focus in Alzheimer's research and clinical practice. It emphasizes the need for timely diagnosis, access to appropriate care, and the development of effective interventions that can delay symptom progression and improve outcomes for individuals with Alzheimer's disease and their caregivers.

**9. FUTURE SCOPE**

Improved Diagnostic Techniques: Advancements in diagnostic techniques, such as biomarkers, neuroimaging, and genetic profiling, may allow for earlier and more accurate detection of Alzheimer's disease. This could enable interventions to be initiated at even earlier stages, potentially enhancing their effectiveness.

Disease-Modifying Therapies: Ongoing research is focused on developing disease-modifying therapies that can target the underlying mechanisms of Alzheimer's disease, aiming to slow or halt disease progression. As these therapies advance, early intervention may become even more crucial to maximize their benefits.

Personalized Treatment Approaches: With a deeper understanding of the heterogeneity of Alzheimer's disease, personalized treatment approaches could be developed. This involves tailoring interventions based on an individual's unique genetic, clinical, and lifestyle characteristics, optimizing their response to treatment.

Digital Health Technologies: The integration of digital health technologies, such as wearable devices, mobile applications, and remote monitoring systems, can support early detection, disease management, and treatment adherence. These technologies have the potential to enhance early intervention by enabling continuous monitoring, personalized interventions, and timely interventions.

Lifestyle Interventions: Lifestyle factors, including physical exercise, cognitive stimulation, diet, and social engagement, have shown promise in reducing the risk of Alzheimer's disease and slowing its progression. Future research can focus on developing evidence-based lifestyle interventions and promoting their adoption at an early stage to optimize brain health.

Supportive Care and Caregiver Interventions: Early intervention can also encompass comprehensive supportive care and interventions targeted at caregivers. Developing programs that provide education, respite care, and psychological support can alleviate the burden on caregivers and enhance the overall care provided to individuals with Alzheimer's disease.

Global Collaboration and Policy Initiatives: The future scope of early intervention also involves global collaboration among researchers, healthcare professionals, policymakers, and advocacy organizations. Collaborative efforts can lead to the development of standardized protocols, guidelines, and policies that prioritize early detection, intervention, and access to appropriate care for individuals with Alzheimer's disease.

**10. APPENDIX**

In [1]:

import tensorflow as tf

import matplotlib.pyplot as plt

import numpy as np

import os

import tensorflow as tf

from tensorflow import keras

from keras import layers

import matplotlib.image as img

import random

import pathlib

%matplotlib inline

In [2]:

path = '../input/alzheimer-mri-dataset/Dataset/'

data\_dir = pathlib.Path(path)

DATA LOAD[¶](https://www.kaggle.com/code/kavinblack/alzheimer-detection-and-classification#DATA-LOAD)

In [3]:

!pip install split-folders

import splitfolders

splitfolders.ratio('../input/alzheimer-mri-dataset/Dataset', output="output", seed=1345, ratio=(.8, 0.1,0.1))

Collecting split-folders

Downloading split\_folders-0.5.1-py3-none-any.whl (8.4 kB)

Installing collected packages: split-folders

Successfully installed split-folders-0.5.1

WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

Copying files: 6400 files [00:44, 144.74 files/s]

In [4]:

IMG\_HEIGHT = 128

IMG\_WIDTH = 128

train\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

"./output/train",

seed=123,

image\_size=(IMG\_HEIGHT, IMG\_WIDTH),

batch\_size=64

)

test\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

"./output/test",

seed=123,

image\_size=(IMG\_HEIGHT, IMG\_WIDTH),

batch\_size=64

)

val\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

"./output/val",

seed=123,

image\_size=(IMG\_HEIGHT, IMG\_WIDTH),

batch\_size=64

)

Found 5119 files belonging to 4 classes.

Found 642 files belonging to 4 classes.

Found 639 files belonging to 4 classes.

In [5]:

class\_names = train\_ds.class\_names

print(class\_names)

train\_ds

['Mild\_Demented', 'Moderate\_Demented', 'Non\_Demented', 'Very\_Mild\_Demented']

Out[5]:

<BatchDataset shapes: ((None, 128, 128, 3), (None,)), types: (tf.float32, tf.int32)>

EXAMPLE IMAGE[¶](https://www.kaggle.com/code/kavinblack/alzheimer-detection-and-classification#EXAMPLE-IMAGE)

In [6]:

plt.figure(figsize=(10, 10))

for images, labels **in** train\_ds.take(1):

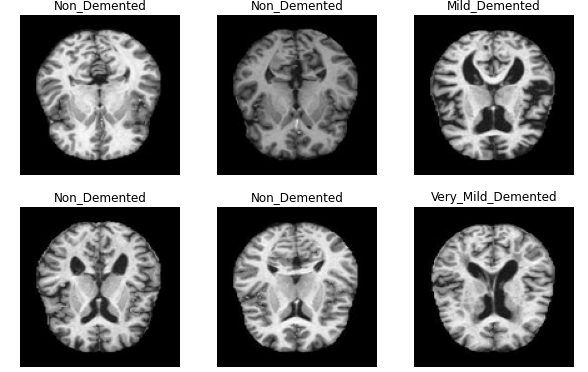
for i **in** range(9):

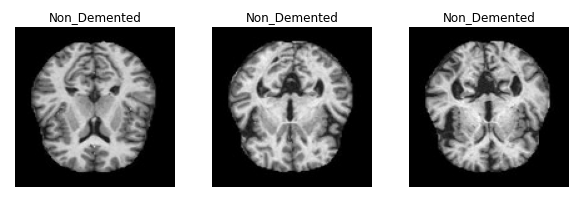
ax = plt.subplot(3, 3, i + 1)

plt.imshow(images[i].numpy().astype("uint8"))

plt.title(class\_names[labels[i]])

plt.axis("off")

****



In [7]:

fig = plt.figure()

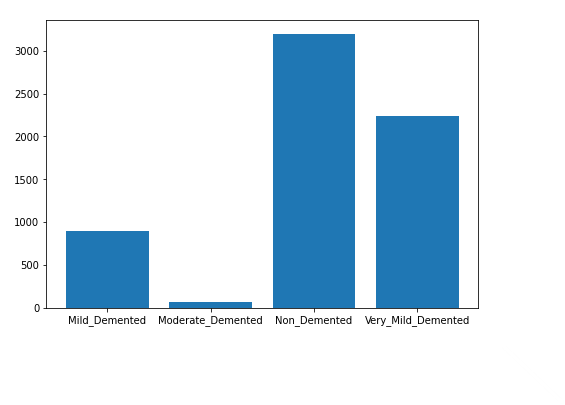
ax = fig.add\_axes([0,0,1,1])

size = [896,64,3200,2240]

ax.bar(class\_names,size)

plt.show

<function matplotlib.pyplot.show(close=None, block=None)>



MODEL[¶](https://www.kaggle.com/code/kavinblack/alzheimer-detection-and-classification#MODEL)

In [8]:

model = keras.models.Sequential()

model.add(keras.layers.experimental.preprocessing.Rescaling(1./255, input\_shape=(IMG\_HEIGHT,IMG\_WIDTH, 3)))

model.add(keras.layers.Conv2D(filters=16,kernel\_size=(3,3),padding='same',activation='relu',kernel\_initializer="he\_normal"))

model.add(keras.layers.MaxPooling2D(pool\_size=(2,2)))

model.add(keras.layers.Conv2D(filters=32,kernel\_size=(3,3),padding='same',activation='relu',kernel\_initializer="he\_normal"))

model.add(keras.layers.MaxPooling2D(pool\_size=(2,2)))

model.add(keras.layers.Dropout(0.20))

model.add(keras.layers.Conv2D(filters=64,kernel\_size=(3,3),padding='same',activation='relu',kernel\_initializer="he\_normal"))

model.add(keras.layers.MaxPooling2D(pool\_size=(2,2)))

model.add(keras.layers.Dropout(0.25))

model.add(keras.layers.Flatten())

model.add(keras.layers.Dense(128,activation="relu",kernel\_initializer="he\_normal"))

model.add(keras.layers.Dense(64,"relu"))

model.add(keras.layers.Dense(4,"softmax"))

In [9]:

model.compile(loss="sparse\_categorical\_crossentropy",

optimizer = "Adam",metrics=["accuracy"])

In [10]:

model.summary()

Model: "sequential"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

rescaling (Rescaling) (None, 128, 128, 3) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d (Conv2D) (None, 128, 128, 16) 448

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 64, 64, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 64, 64, 32) 4640

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 32, 32, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout (Dropout) (None, 32, 32, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 32, 32, 64) 18496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 16, 16, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_1 (Dropout) (None, 16, 16, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 16384) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 128) 2097280

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 64) 8256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 4) 260

=================================================================

Total params: 2,129,380

Trainable params: 2,129,380

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In [11]:

hist = model.fit(train\_ds,validation\_data=val\_ds,epochs=100, batch\_size=64, verbose=1)

Epoch 1/100

80/80 [==============================] - 10s 31ms/step - loss: 1.0005 - accuracy: 0.5357 - val\_loss: 0.8589 - val\_accuracy: 0.6197

Epoch 2/100

80/80 [==============================] - 2s 24ms/step - loss: 0.8062 - accuracy: 0.6509 - val\_loss: 0.6955 - val\_accuracy: 0.7324

Epoch 3/100

80/80 [==============================] - 2s 23ms/step - loss: 0.6918 - accuracy: 0.7040 - val\_loss: 0.5661 - val\_accuracy: 0.7637

Epoch 4/100

80/80 [==============================] - 2s 24ms/step - loss: 0.5604 - accuracy: 0.7609 - val\_loss: 0.5237 - val\_accuracy: 0.8122

Epoch 5/100

80/80 [==============================] - 2s 25ms/step - loss: 0.4455 - accuracy: 0.8134 - val\_loss: 0.4159 - val\_accuracy: 0.8419

Epoch 6/100

80/80 [==============================] - 2s 23ms/step - loss: 0.3330 - accuracy: 0.8746 - val\_loss: 0.2740 - val\_accuracy: 0.8873

Epoch 7/100

80/80 [==============================] - 2s 23ms/step - loss: 0.2479 - accuracy: 0.9039 - val\_loss: 0.4113 - val\_accuracy: 0.8138

Epoch 8/100

80/80 [==============================] - 2s 24ms/step - loss: 0.2151 - accuracy: 0.9187 - val\_loss: 0.1909 - val\_accuracy: 0.9421

Epoch 9/100

80/80 [==============================] - 2s 24ms/step - loss: 0.1397 - accuracy: 0.9496 - val\_loss: 0.1309 - val\_accuracy: 0.9562

Epoch 10/100

80/80 [==============================] - 2s 25ms/step - loss: 0.1115 - accuracy: 0.9611 - val\_loss: 0.0938 - val\_accuracy: 0.9687

Epoch 11/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0864 - accuracy: 0.9697 - val\_loss: 0.0891 - val\_accuracy: 0.9687

Epoch 12/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0938 - accuracy: 0.9662 - val\_loss: 0.0634 - val\_accuracy: 0.9781

Epoch 13/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0868 - accuracy: 0.9682 - val\_loss: 0.0597 - val\_accuracy: 0.9828

Epoch 14/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0681 - accuracy: 0.9746 - val\_loss: 0.0600 - val\_accuracy: 0.9765

Epoch 15/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0579 - accuracy: 0.9789 - val\_loss: 0.0556 - val\_accuracy: 0.9812

Epoch 16/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0437 - accuracy: 0.9865 - val\_loss: 0.0322 - val\_accuracy: 0.9906

Epoch 17/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0564 - accuracy: 0.9787 - val\_loss: 0.0493 - val\_accuracy: 0.9812

Epoch 18/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0486 - accuracy: 0.9826 - val\_loss: 0.0487 - val\_accuracy: 0.9812

Epoch 19/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0384 - accuracy: 0.9887 - val\_loss: 0.0328 - val\_accuracy: 0.9890

Epoch 20/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0475 - accuracy: 0.9836 - val\_loss: 0.1336 - val\_accuracy: 0.9421

Epoch 21/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0392 - accuracy: 0.9875 - val\_loss: 0.0470 - val\_accuracy: 0.9828

Epoch 22/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0364 - accuracy: 0.9861 - val\_loss: 0.0366 - val\_accuracy: 0.9875

Epoch 23/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0190 - accuracy: 0.9916 - val\_loss: 0.0408 - val\_accuracy: 0.9812

Epoch 24/100

80/80 [==============================] - 2s 28ms/step - loss: 0.0353 - accuracy: 0.9875 - val\_loss: 0.0269 - val\_accuracy: 0.9922

Epoch 25/100

80/80 [==============================] - 2s 30ms/step - loss: 0.0320 - accuracy: 0.9900 - val\_loss: 0.0297 - val\_accuracy: 0.9890

Epoch 26/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0239 - accuracy: 0.9926 - val\_loss: 0.0464 - val\_accuracy: 0.9844

Epoch 27/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0565 - accuracy: 0.9807 - val\_loss: 0.0696 - val\_accuracy: 0.9734

Epoch 28/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0321 - accuracy: 0.9889 - val\_loss: 0.0358 - val\_accuracy: 0.9844

Epoch 29/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0288 - accuracy: 0.9898 - val\_loss: 0.0252 - val\_accuracy: 0.9890

Epoch 30/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0278 - accuracy: 0.9896 - val\_loss: 0.0239 - val\_accuracy: 0.9953

Epoch 31/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0302 - accuracy: 0.9877 - val\_loss: 0.0351 - val\_accuracy: 0.9828

Epoch 32/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0178 - accuracy: 0.9941 - val\_loss: 0.0524 - val\_accuracy: 0.9812

Epoch 33/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0208 - accuracy: 0.9928 - val\_loss: 0.0617 - val\_accuracy: 0.9781

Epoch 34/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0329 - accuracy: 0.9883 - val\_loss: 0.0310 - val\_accuracy: 0.9922

Epoch 35/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0142 - accuracy: 0.9965 - val\_loss: 0.0210 - val\_accuracy: 0.9953

Epoch 36/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0243 - accuracy: 0.9930 - val\_loss: 0.0483 - val\_accuracy: 0.9828

Epoch 37/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0113 - accuracy: 0.9959 - val\_loss: 0.0347 - val\_accuracy: 0.9890

Epoch 38/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0150 - accuracy: 0.9953 - val\_loss: 0.0297 - val\_accuracy: 0.9859

Epoch 39/100

80/80 [==============================] - 3s 31ms/step - loss: 0.0273 - accuracy: 0.9891 - val\_loss: 0.0148 - val\_accuracy: 0.9969

Epoch 40/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0116 - accuracy: 0.9955 - val\_loss: 0.0205 - val\_accuracy: 0.9922

Epoch 41/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0380 - accuracy: 0.9891 - val\_loss: 0.0222 - val\_accuracy: 0.9953

Epoch 42/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0249 - accuracy: 0.9906 - val\_loss: 0.0291 - val\_accuracy: 0.9906

Epoch 43/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0131 - accuracy: 0.9963 - val\_loss: 0.0113 - val\_accuracy: 0.9969

Epoch 44/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0196 - accuracy: 0.9937 - val\_loss: 0.0137 - val\_accuracy: 0.9953

Epoch 45/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0078 - accuracy: 0.9982 - val\_loss: 0.0128 - val\_accuracy: 0.9969

Epoch 46/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0169 - accuracy: 0.9937 - val\_loss: 0.0662 - val\_accuracy: 0.9781

Epoch 47/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0225 - accuracy: 0.9924 - val\_loss: 0.0133 - val\_accuracy: 0.9984

Epoch 48/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0166 - accuracy: 0.9947 - val\_loss: 0.0054 - val\_accuracy: 1.0000

Epoch 49/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0154 - accuracy: 0.9949 - val\_loss: 0.0233 - val\_accuracy: 0.9906

Epoch 50/100

80/80 [==============================] - 2s 28ms/step - loss: 0.0113 - accuracy: 0.9953 - val\_loss: 0.0159 - val\_accuracy: 0.9969

Epoch 51/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0052 - accuracy: 0.9988 - val\_loss: 0.0154 - val\_accuracy: 0.9953

Epoch 52/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0163 - accuracy: 0.9945 - val\_loss: 0.0186 - val\_accuracy: 0.9922

Epoch 53/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0218 - accuracy: 0.9922 - val\_loss: 0.0230 - val\_accuracy: 0.9922

Epoch 54/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0140 - accuracy: 0.9959 - val\_loss: 0.0344 - val\_accuracy: 0.9875

Epoch 55/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0124 - accuracy: 0.9945 - val\_loss: 0.0221 - val\_accuracy: 0.9922

Epoch 56/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0154 - accuracy: 0.9953 - val\_loss: 0.0708 - val\_accuracy: 0.9812

Epoch 57/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0289 - accuracy: 0.9904 - val\_loss: 0.0268 - val\_accuracy: 0.9890

Epoch 58/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0165 - accuracy: 0.9934 - val\_loss: 0.0314 - val\_accuracy: 0.9875

Epoch 59/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0176 - accuracy: 0.9941 - val\_loss: 0.0181 - val\_accuracy: 0.9953

Epoch 60/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0196 - accuracy: 0.9937 - val\_loss: 0.0447 - val\_accuracy: 0.9812

Epoch 61/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0167 - accuracy: 0.9937 - val\_loss: 0.0399 - val\_accuracy: 0.9890

Epoch 62/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0135 - accuracy: 0.9945 - val\_loss: 0.0129 - val\_accuracy: 0.9953

Epoch 63/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0212 - accuracy: 0.9930 - val\_loss: 0.0295 - val\_accuracy: 0.9890

Epoch 64/100

80/80 [==============================] - 2s 27ms/step - loss: 0.0116 - accuracy: 0.9949 - val\_loss: 0.0124 - val\_accuracy: 0.9953

Epoch 65/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0176 - accuracy: 0.9947 - val\_loss: 0.0097 - val\_accuracy: 0.9969

Epoch 66/100

80/80 [==============================] - 2s 27ms/step - loss: 0.0171 - accuracy: 0.9936 - val\_loss: 0.0059 - val\_accuracy: 1.0000

Epoch 67/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0079 - accuracy: 0.9977 - val\_loss: 0.0085 - val\_accuracy: 0.9984

Epoch 68/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0122 - accuracy: 0.9957 - val\_loss: 0.0197 - val\_accuracy: 0.9953

Epoch 69/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0101 - accuracy: 0.9967 - val\_loss: 0.0076 - val\_accuracy: 0.9969

Epoch 70/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0123 - accuracy: 0.9955 - val\_loss: 0.0135 - val\_accuracy: 0.9953

Epoch 71/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0224 - accuracy: 0.9910 - val\_loss: 0.0329 - val\_accuracy: 0.9890

Epoch 72/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0124 - accuracy: 0.9963 - val\_loss: 0.0192 - val\_accuracy: 0.9937

Epoch 73/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0096 - accuracy: 0.9959 - val\_loss: 0.0310 - val\_accuracy: 0.9922

Epoch 74/100

80/80 [==============================] - 2s 22ms/step - loss: 0.0083 - accuracy: 0.9967 - val\_loss: 0.0155 - val\_accuracy: 0.9937

Epoch 75/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0220 - accuracy: 0.9930 - val\_loss: 0.0277 - val\_accuracy: 0.9906

Epoch 76/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0073 - accuracy: 0.9975 - val\_loss: 0.0246 - val\_accuracy: 0.9953

Epoch 77/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0156 - accuracy: 0.9955 - val\_loss: 0.0089 - val\_accuracy: 0.9969

Epoch 78/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0119 - accuracy: 0.9957 - val\_loss: 0.0170 - val\_accuracy: 0.9906

Epoch 79/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0099 - accuracy: 0.9971 - val\_loss: 0.0051 - val\_accuracy: 0.9984

Epoch 80/100

80/80 [==============================] - 2s 27ms/step - loss: 0.0065 - accuracy: 0.9973 - val\_loss: 0.0213 - val\_accuracy: 0.9937

Epoch 81/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0188 - accuracy: 0.9945 - val\_loss: 0.0054 - val\_accuracy: 0.9984

Epoch 82/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0027 - accuracy: 0.9996 - val\_loss: 0.0094 - val\_accuracy: 0.9969

Epoch 83/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0071 - accuracy: 0.9980 - val\_loss: 0.0072 - val\_accuracy: 0.9984

Epoch 84/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0049 - accuracy: 0.9984 - val\_loss: 0.0141 - val\_accuracy: 0.9922

Epoch 85/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0078 - accuracy: 0.9986 - val\_loss: 0.0290 - val\_accuracy: 0.9890

Epoch 86/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0141 - accuracy: 0.9951 - val\_loss: 0.0196 - val\_accuracy: 0.9922

Epoch 87/100

80/80 [==============================] - 2s 25ms/step - loss: 0.0103 - accuracy: 0.9969 - val\_loss: 0.0102 - val\_accuracy: 0.9953

Epoch 88/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0241 - accuracy: 0.9906 - val\_loss: 0.0327 - val\_accuracy: 0.9890

Epoch 89/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0321 - accuracy: 0.9898 - val\_loss: 0.0242 - val\_accuracy: 0.9937

Epoch 90/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0096 - accuracy: 0.9971 - val\_loss: 0.0303 - val\_accuracy: 0.9937

Epoch 91/100

80/80 [==============================] - 2s 28ms/step - loss: 0.0053 - accuracy: 0.9980 - val\_loss: 0.0224 - val\_accuracy: 0.9922

Epoch 92/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0065 - accuracy: 0.9977 - val\_loss: 0.0118 - val\_accuracy: 0.9953

Epoch 93/100

80/80 [==============================] - 3s 31ms/step - loss: 0.0070 - accuracy: 0.9977 - val\_loss: 0.0141 - val\_accuracy: 0.9937

Epoch 94/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0131 - accuracy: 0.9945 - val\_loss: 0.0293 - val\_accuracy: 0.9890

Epoch 95/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0216 - accuracy: 0.9930 - val\_loss: 0.0499 - val\_accuracy: 0.9797

Epoch 96/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0101 - accuracy: 0.9969 - val\_loss: 0.0115 - val\_accuracy: 0.9953

Epoch 97/100

80/80 [==============================] - 2s 26ms/step - loss: 0.0063 - accuracy: 0.9980 - val\_loss: 0.0046 - val\_accuracy: 0.9984

Epoch 98/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0085 - accuracy: 0.9977 - val\_loss: 0.0295 - val\_accuracy: 0.9906

Epoch 99/100

80/80 [==============================] - 2s 24ms/step - loss: 0.0037 - accuracy: 0.9984 - val\_loss: 0.0129 - val\_accuracy: 0.9922

Epoch 100/100

80/80 [==============================] - 2s 23ms/step - loss: 0.0111 - accuracy: 0.9965 - val\_loss: 0.0400 - val\_accuracy: 0.9922

Plot the result[¶](https://www.kaggle.com/code/kavinblack/alzheimer-detection-and-classification#Plot-the-result)

In [12]:

get\_ac = hist.history['accuracy']

get\_los = hist.history['loss']

val\_acc = hist.history['val\_accuracy']

val\_loss = hist.history['val\_loss']

In [13]:

epochs = range(len(get\_ac))

plt.plot(epochs, get\_ac, 'g', label='Accuracy of Training data')

plt.plot(epochs, get\_los, 'r', label='Loss of Training data')

plt.title('Training data accuracy and loss')

plt.legend(loc=0)

plt.figure()

plt.plot(epochs, get\_ac, 'g', label='Accuracy of Training Data')

plt.plot(epochs, val\_acc, 'r', label='Accuracy of Validation Data')

plt.title('Training and Validation Accuracy')

plt.legend(loc=0)

plt.figure()

plt.plot(epochs, get\_los, 'g', label='Loss of Training Data')

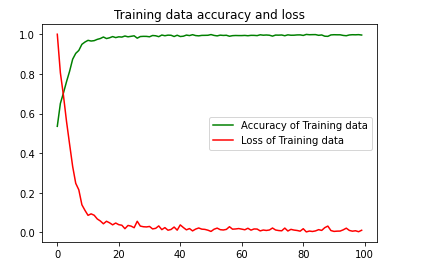
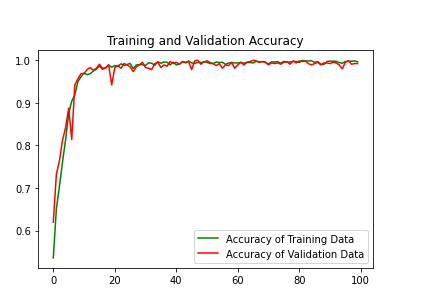
plt.plot(epochs, val\_loss, 'r', label='Loss of Validation Data')

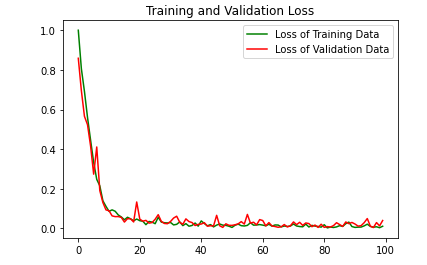
plt.title('Training and Validation Loss')

plt.legend(loc=0)

plt.figure()

plt.show()





<Figure size 432x288 with 0 Axes>

Predictions[¶](https://www.kaggle.com/code/kavinblack/alzheimer-detection-and-classification#Predictions)

In [14]:

def plot(path,class\_name):

print(path)

plt.figure(figsize=(8,8))

img = plt.imread(path)

plt.xticks([])

plt.yticks([])

plt.title("Class Name: "+class\_name)

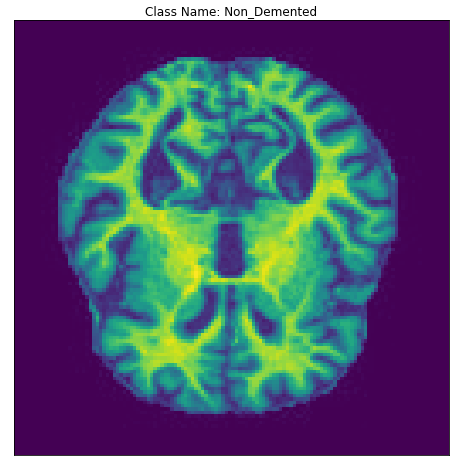
plt.imshow(img)

In [15]:

Very\_Mild\_Demented = random.choice(list(data\_dir.glob("Very\_Mild\_Demented/\*.jpg")))

plot(str(Very\_Mild\_Demented),"Very\_Mild\_Demented")

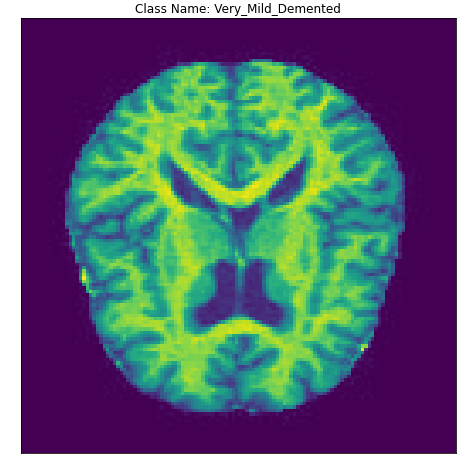
../input/alzheimer-mri-dataset/Dataset/Very\_Mild\_Demented/verymild\_1502.jpg



In [16]:

Mild\_Demented = random.choice(list(data\_dir.glob("Mild\_Demented/\*.jpg")))

plot(str(Mild\_Demented),"Mild\_Demented")

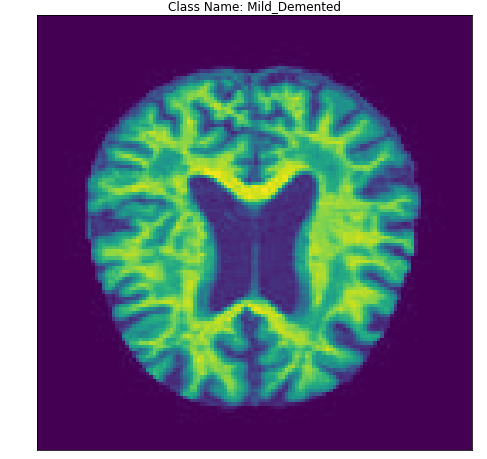


In [17]:

Mild\_Demented = random.choice(list(data\_dir.glob("Mild\_Demented/\*.jpg")))

plot(str(Mild\_Demented),"Mild\_Demented")

../input/alzheimer-mri-dataset/Dataset/Mild\_Demented/mild\_887.jpg

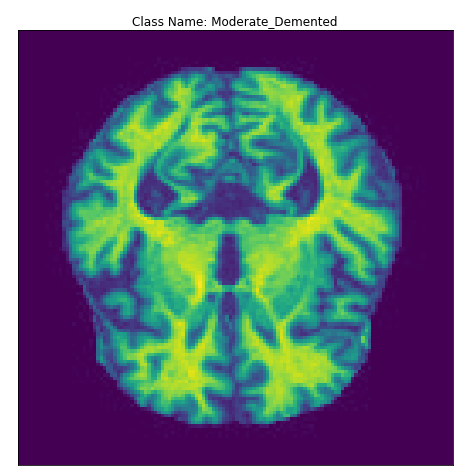


In [18]:

Moderate\_Demented = random.choice(list(data\_dir.glob("Moderate\_Demented/\*.jpg")))

plot(str(Moderate\_Demented),"Moderate\_Demented")

../input/alzheimer-mri-dataset/Dataset/Moderate\_Demented/moderate\_20.jpg



In [19]:

loss, accuracy = model.evaluate(test\_ds)

11/11 [==============================] - 0s 12ms/step - loss: 0.0410 - accuracy: 0.9907

In [20]:

plt.figure(figsize=(20, 20))

for images, labels **in** test\_ds.take(1):

for i **in** range(16):

ax = plt.subplot(4, 4, i + 1)

plt.imshow(images[i].numpy().astype("uint8"))

predictions = model.predict(tf.expand\_dims(images[i], 0))

score = tf.nn.softmax(predictions[0])

if(class\_names[labels[i]]==class\_names[np.argmax(score)]):

plt.title("Actual: "+class\_names[labels[i]])

plt.ylabel("Predicted: "+class\_names[np.argmax(score)],fontdict={'color':'green'})

else:

plt.title("Actual: "+class\_names[labels[i]])

plt.ylabel("Predicted: "+class\_names[np.argmax(score)],fontdict={'color':'red'})

plt.gca().axes.yaxis.set\_ticklabels([])

plt.gca().axes.xaxis.set\_ticklabels([])

