DementiArt: AI-Driven Image Generation for Personalized Memories

***Abstract* –** *DementiArt is an AI-driven web platform designed to generate personalized images from user-provided text descriptions. Leveraging the Stable Diffusion model, the platform transforms textual prompts into visual representations, exploring technology potentially relevant for applications inspired by memory engagement. The system employs a React frontend for user interaction and a FastAPI backend to manage image generation requests via API calls. Cloudinary is utilized for image storage and delivery, while PostgreSQL handles associated metadata. The user interface facilitates the core text-to-image generation workflow. Testing, including unit and integration checks, was conducted to validate the basic functionality and interaction between components. This project successfully demonstrates the integration of these technologies to create a functional, web-accessible AI image generation tool deployed on a scalable cloud infrastructure, providing a foundation for potential future development in user-centric AI applications.*

# Introduction

Dementia is a progressive neurodegenerative condition characterized by a decline in cognitive functions, including memory, thinking, language, and problem-solving. This condition significantly impacts an individual's ability to perform daily activities and maintain independence, posing a growing global health challenge due to the aging population (WHO, 2020). Current treatments primarily focus on managing symptoms rather than reversing the disease's progression, highlighting the need for innovative therapeutic interventions. Engaging with creative and stimulating content can offer cognitive benefits and emotional engagement for individuals with dementia.

This report outlines the design and development of a memory-generator application aimed at providing a simple and engaging tool for individuals, potentially including those with dementia, to interact with AI-generated visuals based on text prompts. The application leverages artificial intelligence technologies, specifically an AI art generation model, to create images from user-provided text. By allowing users to input descriptive text and receive corresponding visual outputs, the application seeks to offer a platform for creative exploration and potential cognitive stimulation. This project aims to provide a user-friendly tool that can be accessed easily, offering a straightforward way to generate visual representations of thoughts and ideas. The development process focuses on simplicity and ease of use.

# Background

Dementia is a progressive neurodegenerative condition that impairs cognitive functions, including memory, reasoning, and communication (Stearn, 2023). As a leading cause of neurocognitive disorder in older adults, dementia presents a significant challenge to global health and healthcare systems. In 2020, the World Health Organization reported over 55 million individuals living with dementia worldwide, with projections indicating a substantial increase in the future (WHO, 2020). Alzheimer’s disease is the most prevalent form of dementia, followed by vascular dementia and frontotemporal dementia (Alzheimer's Society, 2023). The decline in cognitive abilities associated with dementia leads to increasing difficulties in performing daily activities, engaging in social interactions, and managing emotions. This progressive condition also places a considerable burden on caregivers, who often cope with behavioural changes, mood fluctuations, and increasing dependence from the individuals they support. While pharmacological treatments can help manage some symptoms, they do not alter the course of the disease, underscoring the importance of non-pharmacological interventions in dementia care.

Distinguishing between age-related memory changes and dementia involves evaluating aspects such as vision, mood, and decision-making (Stearn, 2023). Experts note that while occasional forgetfulness can occur with age, persistent cognitive decline that affects an individual's daily life may be indicative of dementia. A definitive diagnosis, however, requires assessment by a qualified medical professional (Stearn, 2023).

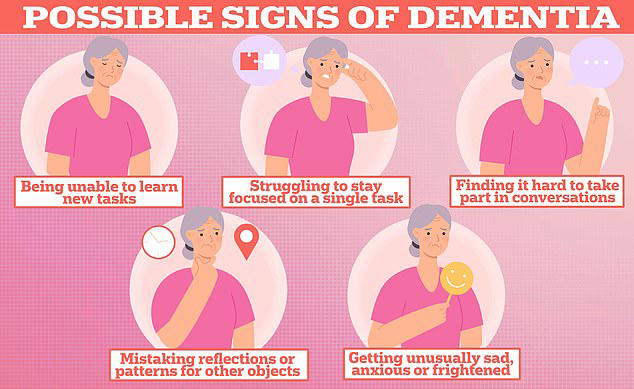


Figure : Possible Signs of Dementia (Stearn, 2023).

Engaging with visual content and creative activities can potentially offer cognitive stimulation and emotional engagement for individuals. AI-driven art generation is an area of growing interest, providing tools that can create images based on textual input (Liu et al., 2021; Zhou et al., 2023). These systems use algorithms to interpret text prompts and generate corresponding visual outputs. Interaction with such tools may offer a simple and accessible form of engagement for a wide range of users.

The healthcare sector is increasingly exploring the use of technology to develop systems that aim to support individuals with various needs. User-friendly applications that provide simple interfaces and clear functionalities can be beneficial for those who may have difficulties navigating complex systems.

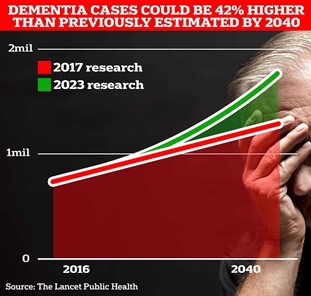


Figure : Growth in Estimated Cases of Dementia (2017-2040) (Stearn, 2023)

Research suggests that the prevalence of dementia is projected to increase beyond previous estimates. Stearn (2023) reports that dementia cases are anticipated to be 42% higher by 2040 compared to 2017 projections. The significant increase in estimated dementia prevalence is shown in the green curve (2023 research), which exceeds the red curve (2017 research). Factors contributing to this increasing incidence include an aging population, the prevalence of certain lifestyle conditions, and advancements in diagnostic accuracy. These findings highlight the growing need for accessible and user-friendly tools that can provide engagement and potential cognitive benefits.

# Specification

**3.1 Defining the Problem and System Requirements**

Individuals may benefit from simple and accessible tools that allow them to engage with creative visual content. There is a need for applications that offer a straightforward way to generate images from text prompts, providing a medium for creative expression and potential cognitive stimulation.

The memory-generator application aims to address this need by providing a user-friendly interface where individuals can input text and receive corresponding AI-generated images. The system should be easily accessible via web browsers and offer a simple mechanism for generating visuals. Key requirements include:

**Text Input:** Users should be able to easily input text prompts.

**Image Generation:** The system must be capable of generating relevant images based on the provided text prompts using an AI model.

**Image Display:** The generated images should be clearly displayed to the user.

**User-Friendly Interface:** The interface should be intuitive and easy to navigate for a wide range of users.

**Accessibility:** The application should be accessible through standard web browsers.

**3.2 Identifying Stakeholders and User Needs**

The primary users of this memory-generator application are individuals who are interested in generating visual content from text prompts. This could include a broad range of users seeking creative engagement or a simple tool for visual representation. While the initial description focused on dementia patients, the current functionality suggests a wider user base. Caregivers or family members might also use the application to create visual aids or engaging content for individuals they support. The key needs of the users include simplicity, ease of use, and the ability to quickly generate relevant images from their text inputs.

**3.3 Developing the System Requirements**

The memory-generator application requires a well-defined set of functional and non-functional specifications to ensure its usability and effectiveness.

**Functional Requirements:**

**FR1:** Text Prompt Input: The system shall provide a clear and easily accessible text field for users to input their prompts.

**FR2:** Image Generation: Upon submission of a text prompt, the system shall use an AI model to generate a corresponding image.

**FR3:** Image Display: The generated image shall be displayed to the user in a clear and viewable format.

**FR4:** History (Optional): The system may optionally store a history of previously generated images for the user.

**Non-Functional Requirements:**

**NFR1:** Performance: The system should generate images within a reasonable timeframe (e.g., within a few seconds).

**NFR2:** Usability: The user interface should be intuitive and easy to navigate for users with varying levels of technical expertise.

**NFR3:** Accessibility: The application should be accessible through standard web browsers on various devices.

**NFR4:** Scalability: The system should be able to handle a reasonable number of concurrent users and requests.

**3.4 Establishing a Work Schedule and Deliverables**

The development of the memory-generator application will follow a structured timeline with key phases:

**Phase 1: Research and Planning:** This phase involves identifying the appropriate AI model for image generation, planning the system architecture, and designing the user interface.

**Phase 2: Development:** This phase focuses on building the frontend and backend components of the application and integrating the AI model.

**Phase 3: Testing:** Thorough testing will be conducted to ensure the application functions correctly and meets the specified requirements.

**Phase 4: Deployment:** The application will be deployed to a web hosting platform for user access.

Key deliverables will include the functioning web application, potentially user documentation (depending on complexity), and a report outlining the development process.

# Design

## Introduction to the Design Approach

The DementiArt App is designed with a primary focus on simplicity and ease of use, enabling users to effortlessly create images from textual prompts. The application adopts a web-based approach to ensure broad accessibility across various devices. The core design principle centres around a streamlined process where users input their creative ideas as text, and the system, leveraging artificial intelligence, transforms these ideas into visual representations. The frontend of the application is developed using React, providing a dynamic and responsive user interface. The backend, responsible for processing user requests and interacting with the AI model and image storage, is deployed on the Railway platform. Cloudinary is integrated into the design for efficient storage and delivery of the generated images. This design approach aims to provide a user-friendly and efficient experience for generating visual content from text.

## System Architecture

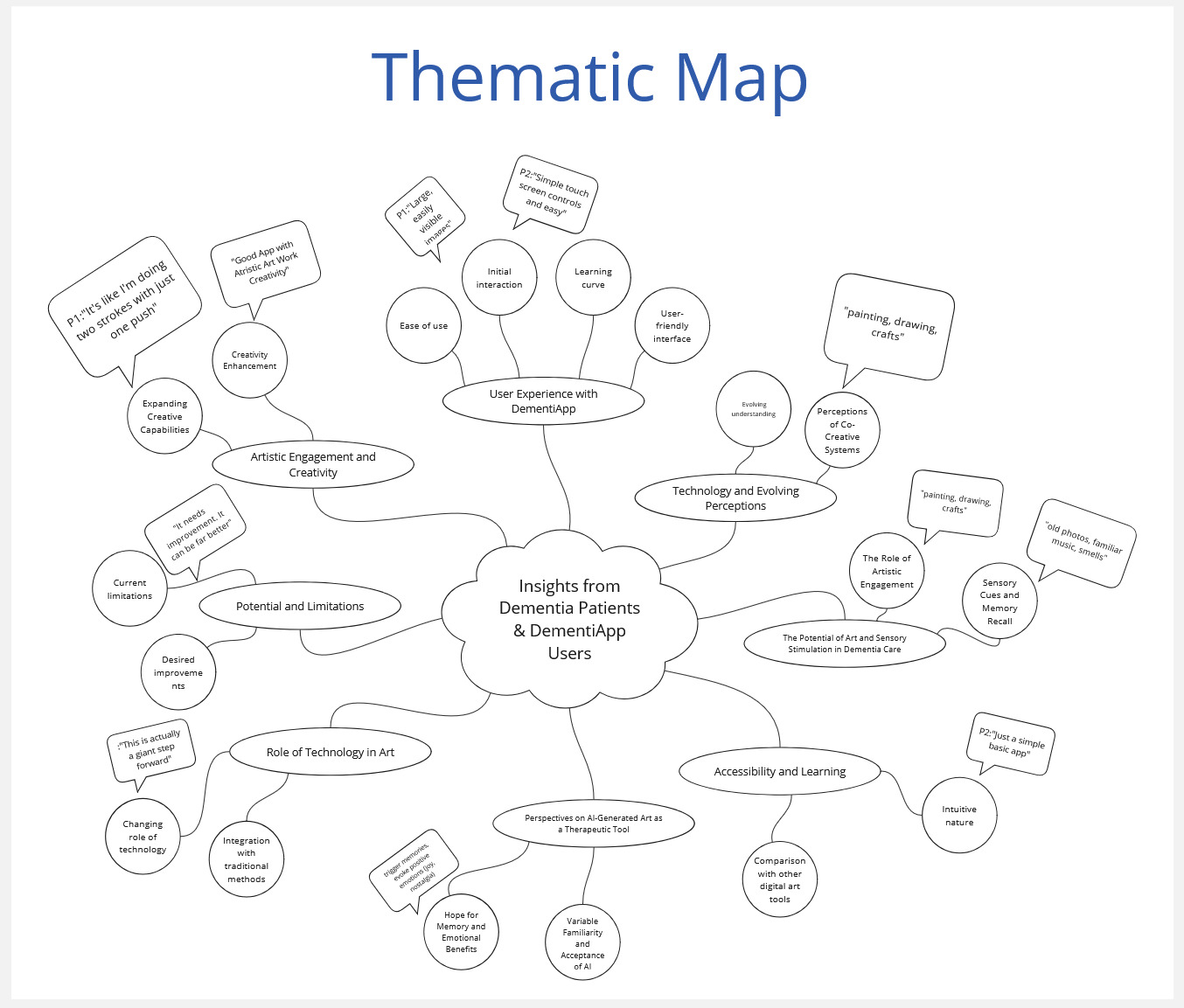
### The DementiArt App follows a client-server architectural pattern. The frontend, built with React, serves as the client-side interface for user interaction. The backend, hosted on Railway, acts as the server, handling user requests, managing the AI model for image generation, and utilizing Cloudinary for image storage.

**4.2.1 Frontend (React)**

The user interface of the **DementiArt App** is developed using React, a JavaScript library known for building interactive and dynamic web applications. The frontend provides users with:

* A clearly labelled text input field for entering their desired prompts.
* An intuitive button to initiate the image generation process.
* A designated area where the generated image is displayed to the user.

The React frontend is responsible for capturing the user's text input and sending it to the backend API. Upon receiving the URL of the generated image from the backend, it renders the image in the display area.



**Figure 3:** Thematic Map

## System Architecture

### The DementiArt App follows a client-server architectural pattern. The frontend, built with React, serves as the client-side interface for user interaction. The backend, hosted on Railway, acts as the server, handling user requests, managing the AI model for image generation, and utilizing Cloudinary for image storage.

**4.2.1 Frontend (React)**

The user interface of the **DementiArt App** is developed using React, a JavaScript library known for building interactive and dynamic web applications. The frontend provides users with:

* A clearly labelled text input field for entering their desired prompts.
* An intuitive button to initiate the image generation process.
* A designated area where the generated image is displayed to the user.

The React frontend is responsible for capturing the user's text input and sending it to the backend API. Upon receiving the URL of the generated image from the backend, it renders the image in the display area.

**4.2.2 Backend (FastAPI)**

The backend of the DementiArt App, deployed on the Railway platform, manages the core logic of the system. It is implemented using FastAPI, which is used to build the RESTful APIs that handle communication with the React frontend and manage user authentication. The backend performs the following key functions:

* Receives text prompts from the frontend via API requests.
* Communicates with a cloud-based text-to-image AI model (via its API) to generate an image based on the received prompt.
* Upon receiving the generated image data from the AI model, it uploads the image to Cloudinary for storage and efficient delivery.
* Retrieves the URL of the stored image from Cloudinary.
* Sends the Cloudinary URL back to the frontend, allowing the image to be displayed to the user.

**4.2.3 AI Model (Likely a cloud-based text-to-image model)**

The **DementiArt App** leverages a pre-trained text-to-image AI model hosted on a cloud platform. This could be a service provided by companies like OpenAI (DALL-E API), Stability AI (Stable Diffusion API), or other similar providers. The backend interacts with this AI model through its API, sending the user-provided text prompt and receiving the generated image data in response.

**4.2.4 Cloud Image Storage (Cloudinary)**

Cloudinary is integrated into the **DementiArt App** as the primary solution for storing and delivering the generated images. Once the backend receives an image from the AI model, it uploads the image to Cloudinary. Cloudinary offers features such as image optimization, resizing, and a global Content Delivery Network (CDN), ensuring that images are delivered efficiently to the user's browser. The backend then uses the URL provided by Cloudinary to send the image location to the frontend.

**4.2.5 Deployment (Railway)**

The **DementiArt App,** encompassing both the React frontend and the backend API, is deployed on the Railway platform. Railway provides a cloud-based infrastructure that simplifies the deployment and management of web applications directly from Git repositories. This platform handles aspects such as server provisioning, scaling, and continuous deployment, allowing developers to focus on the application's functionality.

**4.3 Web Application and Deployment**

The **DementiArt App** is accessible to users through any modern web browser. The deployment on Railway ensures that the application is readily available over the internet.

**4.3.1 Deployment Workflow**

The process for a user to generate an image using the **DementiArt App** involves the following steps:

1. **Access:** The user opens the web application in their browser by navigating to the URL hosted on Railway.
2. **Input:** The user enters a descriptive text prompt into the input field provided by the React frontend.
3. **Request:** The user clicks the "Generate" button (or a similar call to action) on the frontend.
4. **Transmission:** The React frontend sends an API request containing the text prompt to the backend server hosted on Railway.
5. **AI Interaction:** The backend server receives the prompt and communicates with the chosen cloud-based AI model's API, forwarding the text for image generation.
6. **Generation:** The AI model processes the text and generates a corresponding image.
7. **Storage:** The backend server receives the generated image data and uploads it to Cloudinary.
8. **URL Retrieval:** Cloudinary stores the image and returns a unique URL to the backend server.
9. **Response:** The backend server sends the Cloudinary URL back to the user's browser (React frontend).
10. **Display:** The React frontend receives the image URL and renders the image in the designated display area of the user interface.

**4.3.2 System Architecture and Deployment**

The **DementiArt App** utilizes a robust and scalable cloud-based architecture, fully deployed on the Railway platform. The key components work together to provide a seamless user experience:

* **React Frontend:** Provides an interactive interface for users to input prompts and view generated images. Hosted on Railway.
* **Backend API:** Manages the application logic, including communication with the AI model and Cloudinary. Likely built with FastAPI and hosted on Railway.
* **Cloud-Based AI Model:** Performs the task of generating images from text prompts, accessed via API.
* **Cloudinary:** Provides secure and efficient storage and delivery of the generated image assets.

The deployment on Railway simplifies the infrastructure management and allows for easy scaling of the application as the user base grows. The integration with Cloudinary ensures efficient handling of image resources, and the use of a cloud-based AI model provides access to powerful image generation capabilities.

## Web Application and Deployment

Users gain a secure web-based dementia care system connection through their smartphones, tablets, and desktop computers. User frontends obtain their functionality from client-server architecture in the system arrangement structure. The platform allows users to access AI content effortlessly, and servers complete all computational operations to ensure that users experience fast service. The FastAPI framework is the backend choice because it provides both lightweight, flexible, and scalable characteristics. Combining sophisticated AI models with frontend UI components produces real-time system efficiency that enables image processing and caption creation functions. The system deployment method utilizes scalable design principles coupled with efficiency standards and user-friendly principles to maintain operational support for dementia patients and their care providers in a supportive operational condition.

### Deployment Workflow

The DementiArt App allows users to generate images based on custom text prompts and view previously generated images. The deployment workflow involves the following key steps:

* **User Interaction:** Upon opening the app, users are presented with a login and then home screen featuring two primary options: "Create a New Memory" and "Play Old Memories.".
* **User Input:** When a user selects "Create a New Memory," they are directed to a screen with a text field to input their desired prompt. This prompt can be a description of various subjects or concepts. The text prompt serves as the input for the image generation process.
* **Image Creation:** Once the user enters a text prompt and clicks the "Generate" button, the app utilizes the Stable Diffusion model to generate an image based on the provided text. This AI model interprets the text prompt and produces a visual representation.
* **Image Staging:** After the image is generated, it is securely stored on Cloudinary, a cloud-based image storage service. Cloudinary facilitates the storage and retrieval of images for display within the application.
* **Backend Processing (FastAPI):** The backend logic is managed by FastAPI, which handles the API requests. When a user submits a prompt, FastAPI receives it and communicates with the Stable Diffusion model to generate the image. The backend also interacts with Cloudinary to store and retrieve the generated images.
* **Memory Visualization:** The generated image is displayed to the user on the frontend, which is built using React. The React interface presents the image below the prompt input area. If the user selects "Play Old Memories," a grid of previously generated images is displayed from the application's storage.
* **Final Output:** Users can view the generated image on the screen after it's created. They also have the option to view a collection of their previously generated images by navigating to the "Play Old Memories" section. The current version of the application does not offer features for downloading, sharing, categorizing into albums, or advanced manipulation of the generated images directly within the app.
* **Railway Deployment**: The entire application, including the frontend and backend, is deployed on Railway, a cloud platform that provides infrastructure for scalability and performance. Railway ensures that DementiArt App is accessible and performs efficiently.

### System Architecture and Deployment

The DementiArt App's novel cloud-based modelling helps in efficient operation and fast response times. The system is designed with a FastAPI-based backend, a React-based front end, and cloud-hosted AI models for image generation. The deployment process ensures perfect link-up between these elements, resulting in a strong and agile application.

* Backend Operations (FastAPI): The backend is built on FastAPI, a modern framework that supports API requests and back-end processing. It enables the fast processing of user requests, handles user data, and links to cloud services such as Cloudinary for images. FastAPI handles real-time communication between the Stable DiffusionAI model and generates images based on user input.
* Frontend Interface (React): The front is created using React, which provides a dynamic, interactive, and user-friendly interface. The interface is simple, where users can input prompts and view the generated outputs. The design is about accessibility, enabling people to navigate and engage in their cherished memories efficiently.
* Cloud-Based AI Models (Stable Diffusion): Stable Diffusion AI models reside on cloud server infrastructures, providing real-time image generation based on user input. This cloud-based architecture reduces the dependency on local devices and results in improved scalability and performance.
* Cloud Image Storage (Cloudinary): The generated images are stored securely using Cloudinary, which has fast and optimized access and storage. This allows users to view, download, and share pictures anytime.
* Safe Data Processing and Storage: The system safeguards users' privacy by collecting user data in a secure way. User information and generated images are stored inside a PostgreSQL database on the backend. Jpeg files are uploaded and processed as soon as they hit S3 and stored on Cloudinary for safe access. Data privacy and security are ensured via confidential connections and respect for appropriate privacy regulations.
* RESTful API Integration: The front and backend are connected through RESTful API endpoints, and fast data transfer between the React-based and FastAPI backend is possible. This integration ensures real-time functionality and scalability and facilitates straightforward communication between user information input, image creation, and image storage processes.

DementiArt App enables users to engage with it by effortlessly coupling its front-end, back-end, and AI models, making for smooth operation and a user-friendly experience. The cloud-based architecture and deployment on the railway scale ensure that the system will be seamless with more users.

A diagram of a memory application

AI-generated content may be incorrect.

**Figure 4:** Deployment Flow of Dementi App

### Scalability and Cloud-Based Processing

DementiArt App utilizes a cloud-based architecture designed for efficient processing and potential scalability. The system employs a modern technology stack, integrating cloud infrastructure to support its functionality. Below are the main points regarding its cloud processing capabilities:

**Backend in Python and FastAPI:** The backend of DementiArt App is implemented in Python using the FastAPI framework to handle API requests. FastAPI is known for its performance and ability to handle concurrent requests efficiently, which is beneficial for processing user prompts and managing image generation.

**Stable Diffusion for Image Synthesis:** DementiArt App uses the Stable Diffusion model, a cloud-based AI model, to generate images based on user-provided text prompts. By leveraging cloud infrastructure for this computationally intensive task, the application can handle image generation without placing a significant load on the user's device.

**Cloudinary for Image Storage:** Images generated by the application are stored and managed using Cloudinary, a cloud-based image storage and management platform. Cloudinary provides a scalable and secure way to store images, ensuring they can be readily accessed for display within the application. Its Content Delivery Network (CDN) also contributes to faster loading times for users.

**PostgreSQL for Data Storage:** While PostgreSQL is used as the database for the application, the current live version of DementiArt App does not appear to implement user accounts or the storage of user-specific preferences. The database likely stores metadata related to the generated images.

**API Communication:** The backend of DementiArt App uses API calls to interact with external services such as the Stable Diffusion model and Cloudinary for image generation and storage. This allows for a modular design and efficient utilization of specialized cloud services.

**Frontend with React:** The frontend of DementiArt App is built using the React library, which is known for creating interactive and responsive user interfaces. React enables users to input prompts and view the generated images within a dynamic web environment.

**Cloud-Based Infrastructure:** The entire application, including the frontend and backend, is hosted on cloud platforms like Railway. This cloud infrastructure provides the necessary resources for the application to be accessible over the internet and offers the potential for scaling resources as needed.

**Performance Considerations:** By utilizing cloud-based resources for key tasks like image generation and storage, DementiArt App aims to provide a responsive user experience. The cloud architecture helps to minimize latency and optimize processing times for image generation and retrieval.

In summary, DementiArt App leverages cloud services and a modern technology stack including Python, FastAPI, Stable Diffusion, Cloudinary, and React to provide its image generation and viewing functionalities. This architecture supports efficient processing and offers the potential for scalability in the future..

# Implementation and Testing

The implementation stage of the DementiArt App concerns ensuring that the essential components of the app are successfully integrated, working correctly, and ready for launch in the market. This section describes the way of the frontend, backend, and DB/Storage implementation and a brief overview of the testing regime used to make the product reliable, scalable, and fast.

## Introduction to Implementation

DementiArt App’s development process included building the app’s main components with the help of a robust tech stack. Both frontend, backend, and database were built simultaneously to smooth the interface. The back end was created to deal with image creation requests and handle user data, and the front end was designed to give a soft and interactive user experience. Lastly, storage and database solutions were picked to ensure efficient and secure treatment of persons' details and generated pix. The implementation of the following significant components is described below.

### Frontend Implementation

The DementiArt App front is created by React, which provides a lively and dynamic user interface. The creation concentrated on allowing users to seamlessly interact with the app to enter prompts, look at the generated images, and manage their memories.

* UI/UX Design: The interface of the front end has been made user-friendly so that user prompts can be inputted easily. A clean and contemporary design made it simple for owners to look at and touch the images generated.
* State Management: Reac’s state management was used for dynamic data management, like saving generated images and user inputs. This way, images are updated in real-time without reloading the page; it is a great experience.
* Image Display: The image can be retrieved from Cloudinary and shown to the user once it has been generated as an image. The image gallery format allows users to collect and browse through the previously generated content. The front end is also optimized for mobile to support all mobile platforms.
* API Integration: The frontend communicates with the backend via RESTful APIs defined with FastAPI. These APIs handle user input, start image generation, and store image retrieval. The front end ensures that API calls are made asynchronously for better performance.

### Backend Implementation

The backend of the DementiArt App is built with Python and FastAPI for a solid structure to deal with the API requests and backend processing. The backend deals with user question processing, data storing, and cloud services applications for image generation and storage.

* Image Generation: The backend is linked with the Stable Diffusion model to generate images from user prompts. When a user scrolls in with prompt information, the backend sends the input to the AI model with a generated image back. The backend takes care of the communication with the model to be efficient and to respond quickly to the user.

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 5:** Code for Prompt Construction and API Interaction

* API Requests: Both the backend uses FastAPI to manage restful API requests, and it gives a strong ability to manage multiple users with concurrent rates at that time. The FastAPI asynchronous capabilities enable the backend to operate concurrently on several image generation tasks, optimizing resources.
* Security and Authentication: User authentication is a priority on the backend. JWT (JSON Web Tokens) are used to handle login sessions and protect user data.

### Database and Storage

The DementiArt App runs on a cloud-based storage platform and a structured database to organize user data and images produced securely.

* Database Implementation (PostgreSQL): These include the user data in a PostgreSQL database, including login credentials and loading preferences. PostgreSQL is a solid, scalable option for structuring transcription. The data structure in database design allows user information to be arranged neatly and easily obtained, enabling the user to access the created memories easily. Security measures include secured connections, proper data checking, and shielding client information and privacy.
* Image Storage (Cloudinary): Rendered images are stored in Cloudinary, a cloud-based picture management. Cloudinary hosts images securely and delivers them efficiently through its CDN. Images can be downloaded without delay after an upload request and reply to users of the front end. Cloudinary’s infrastructure is scalable, making images accessible during peak traffic times and ensuring performance and reliability.

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 6:** Database Connection Configuration with SQLAlchemy

### Testing Strategy

Testing is an integral part of the implementation process to confirm that all elements of the DementiArt App should behave as required. The approach document for the testing strategy includes several essential techniques:

* Unit Testing: Unit tests are performed on every single item, including surveillance of the posterior from the back end, fore-end, and database. These tests verify that all functions or methods operate as they should, whether creating an image, storing data, or displaying content.
* Integration Testing: Integration Testing is accomplished to see how many different app parts work together. For example, interaction with frontend/backend API and storage/retrieval of images from Cloudinary is quite in-depth.
* Load Testing: Load testing simulates a full traffic load and ensures the DementiArt App can throughput a high user load. This keeps the application responsive to users even if other users are running the application simultaneously.
* User Acceptance Testing (UAT) – Users do it to ensure that the means are employed smoothly, obviously, and without bugs. UAT feedback is utilized to make any last changes to the app before its deployment.

## Web Application Development and Deployment

The building and deployment of the DementiArt App was about combining innovative technologies to make a smooth, fast, large-scale web application. The aim was to give users an interactive interface to produce images from text prompts. This chapter covers the development lifecycle and deployment plan that got us here, focusing on the top technologies used.

### Development Workflow

Construction of the DementiArt App was a structured process that held both frontend and backend participants for loads, working smoothly between them. The front end is constructed on top of React, offering a proactive, reacting user interface that enables users to ask prompts, visualize random images, and control their computing memory. The React-rooted interface leverages state management for real-time content updates, like showing freshly produced images, without the need to refresh this webpage. Based on Python and FastAPI, the backend responds to APIs and deals with user data input. When a user enters a prompt, the backend serves it to the Stable Diffusion AI model, which produces an image upon the prompt command. FastAPI’s async support sends the backend to handle a vast number of incoming requests in parallel efficiently, improving average performance. The backend also communicates with the Postgre database, storing, in it, user information, login credentials, and preferences safely. Images are managed from Cloudinary for storage. Once one image is rendered, it gets uploaded to Cloudinary, where it is Optimized and sent out to users over its Globe Covered CDN. This cloud-based hosting allows images to be accessed rapidly, even under prospective high access.

### Deployment Strategy:

The deployment strategy for DementiArt App aims for scalability and reliability to ensure smooth operation even with a significant user base. The application utilizes cloud technologies for hosting and processing.

The frontend, built with React, is hosted on the cloud platform Railway. This configuration provides users with fast access and low latency from various geographical locations.

The backend, implemented using the FastAPI framework, runs on a cloud service provided by Railway. This allows the backend to scale its resources based on the application's usage. FastAPI's asynchronous execution capabilities enable it to handle multiple API requests concurrently, contributing to a better user experience, even under higher load.

While PostgreSQL is used as the database for the application, the current deployed version of DementiArt App does not appear to have user accounts or store individual user data such as login credentials. PostgreSQL likely stores metadata related to the generated images and other application-specific data. The cloud-hosted PostgreSQL instance offers high availability and data redundancy.

Quick and secure management of user-generated images is achieved through the use of Cloudinary for image storage. Cloudinary's global Content Delivery Network (CDN) ensures that images are readily accessible to users with low latency.

The cloud infrastructure provided by Railway offers auto-scaling capabilities, allowing the backend to automatically adjust its resources to meet fluctuating user traffic. Additionally, load balancing is likely employed to distribute incoming traffic evenly across multiple servers, optimizing performance and preventing any single server from becoming overloaded.

In summary, the deployment strategy for DentiArt App leverages the scalability and reliability of cloud platforms like Railway, along with services such as FastAPI, Stable Diffusion (for image generation), Cloudinary, and PostgreSQL, to provide a responsive and accessible application.

### Performance Optimization

During the development and deployment, numerous techniques were employed to improve performance. With the help of cloud-based AI models like Stable Diffusion, image generation is completed efficiently without needing local processing, generating fast and high-quality results. The only Cloudinary issue is the use of storage images on the cloud. It has an optimized delivery cache, and the image compression will reduce application latency. Cloud infrastructure enables seamless efficiency, ensuring that as many more customers interact with the app, the method can efficiently change dynamically in response to requirements. Caching also improves performance by speeding up retrieval of commonly used data, like images already generated.

### Continuous Deployment and Monitoring

A continuous delivery pipeline is implemented to ensure that the DementiArt App remains up-to-date and stable. This enables automatic deployment of updates and facilitates the rapid deployment of bug fixes. Real-time monitoring tools are utilized to inspect the application's functionality and identify potential issues before they impact users. Proactive measures, such as automated testing and infrastructure redundancy, are in place to prevent failures and ensure high availability.

## Testing and Debugging

Testing and debugging are key to the efficient, secure, and reliable operation of the DementiArt App. With the app’s various components statement: frontend, backend, image generation, and cloud-based services, every prerequisite using it was to engage a thorough test approach to look for and rectify issues ahead of launch. The following section covers the testing and debugging, with leading technologies and procedures to ensure high-quality standards.

### Unit Testing

So that every part of the application functions as expected, unit testing was implemented across the front-end, the back-end, and image generation components. React components were subjected to tests on the front end that confirmed whether UI elements performed as expected regarding state management, rendering, and user interaction. Unit tests were created for every React component to ensure that user inputs, image display, and memory management items worked correctly. FastAPI endpoints were thoroughly exercised on the backend to ensure they processed the API requests. These tests tested response times, data validation, and proper interaction with the Stable Diffusion model, PostgreSQL database, and Cloudinary. The main objective was to verify that the back, multiple user requirements are supported without due issues and that image generation and data management are well. The stable diffusion model also got unit tests to ensure the image generation function was performed correctly with any user input. These tests confirmed the model’s ability to create high-quality, accurate images from the text prompts.

### Integration Testing

Once all the individual parts were unit-tested, the integration test was carried out to guarantee that the system worked smoothly. This included testing the interaction between front end and back end and integration with cloud services like Cloudinary and PostgreSQL. Communication between the React front end and the FastAPI backend was ensured to verify that API calls in the front end are processed correctly by the backend. This also involved checking that the backend could fetch generated images from Cloudinary and display them and that the user’s data is correctly saved and retrieved from the PostgreSQL database. Moreover, the backend integration with the Stable Diffusion model was validated to check how image generation requests will flow and be processed correctly. Integration tests checked whether the uploaded images were correctly stored at Cloudinary and displayed on the front end for the user.

### Performance Testing

As the DementiArt App is scalable, performance testing was a priority to ensure the app will support the growing user load and maintain user-friendliness, whatever the load. Load testing tools were employed to simulate heavy traffic, sending many concurrent requests, image generation, and login user data retrieval. Performance tests measured the response times for API calls, image rendering times, and PostgreSQL and Cloudinary data storage and retrieval efficiency. The aim was for the system to be able to deal with a large number of users minus slowdowns and crashes. Auto-scaling was also experimented to see that cloud infrastructure can automatically adjust to more traffic without going down.

### Security Testing

Considering the significance of user data privacy, security testing was performed to guarantee that all sensitive information, including login credentials and user preferences, was handled securely. PostgreSQL validated that data storage and retrieval operations were encrypted and that data protection against unauthorized access was ensured. The FastAPI secondary backend also underwent security tests to guarantee that authentication mechanisms (like JWT tokens ) are working well and that Unauthorized access to user's data is blocked. The system was also checked for vulnerabilities in SQL injection and cross-site scripting (XSS) to guarantee the safety and security of the app.

### Debugging Process

The charging-cycle protocol finds and remedies the issue discovered during a debugging process. When bugs were introduced, the development team used logging and error-tracking tools to log its behavior when the app was developed and tested. They allowed us to track exact issues, like API calls, image-generating fails, or database connectivity fails. React Developer Tools were used to debug the frontend issues, and developers could inspect the state of React components and determine any rendering or state management problems. On the backend side, FastAPI offered excellent error messages and logs, which allowed us to determine the issues with API request processing and integration with external services such as Cloudinary and Stable Diffusion.

When mistakes came up the development group concentrated on solving them through a strategy, debugging one tier at any given time—from frontend UI issues to backend processing and also cloud solution interactions; after bugs were fixed, the system was re-tested to make sure that the fixes were not required and that the software was perfect.

### User Acceptance Testing (UAT)

Integration and performance testing were completed after the unit testing phase. Following this, User Acceptance Testing (UAT) was undertaken with a total of five participants. These participants were recruited through personal contacts and consisted of individuals with varying levels of technical experience. The group included three individuals in the age range of 25-40 who regularly use web-based applications and two individuals in the age range of 50-65 with moderate experience using online tools.

During the UAT session, participants were asked to navigate to the DementiArt App homepage and enter at least three different text prompts of their choosing. They were encouraged to experiment with different types of prompts, ranging from simple nouns to more complex descriptive phrases. For each generated image, participants were asked to assess its relevance to their prompt, its visual quality, and the speed of generation. Following their interaction with the app, participants completed a short online questionnaire that asked for feedback on the app's ease of use, the intuitiveness of the interface, and any bugs or areas for improvement they noticed.

Feedback from the users indicated that the app was generally easy to understand and use. However, minor enhancements were suggested regarding the placement of the 'Generate' button and the clarity of the loading indicator. No critical bugs were identified during the UAT. Participants tested the app on both desktop computers and mobile phones, and the feedback indicated that the app was responsive and accessible across multiple devices.

## Challenges and Workarounds

The development and deployment of DementiArt App presented several technical challenges related to the application's functionality and the integration of various technologies. These challenges were addressed through specific workarounds to ensure the application's smooth operation. This section outlines the major issues encountered and the implemented solutions.

**Table 1:** Challenges and Workarounds in System Implementation

| Challenge | Workaround |
| --- | --- |
| Handling Large Image Generation Loads | Image generation was offloaded to a cloud infrastructure to utilize its scalable computational resources. The backend, using FastAPI, implemented asynchronous capabilities to manage multiple concurrent image generation requests. This approach aimed to prevent system slowdowns and maintain application responsiveness. |
| Ensuring Scalability with Cloud-Based Services | Cloudinary was used for image storage and delivery via its global Content Delivery Network (CDN), optimizing image handling and reducing latency. PostgreSQL database performance was improved by using indexing and partitioning techniques for efficient management of increasing user data. Backend auto-scaling was configured to dynamically adjust server resources based on demand. |
| Data Synchronization Between Frontend and Backend | RESTful APIs were implemented to manage data exchange between the React frontend and the FastAPI backend. Asynchronous communication was employed to facilitate timely updates and minimize delays in retrieving and displaying images, contributing to a seamless user experience. |
| Managing Security and User Privacy | JWT tokens were implemented for secure user authentication and session management. All API communication used encrypted HTTPS protocol. Sensitive user data in the PostgreSQL database was encrypted. The application was also tested for common web security vulnerabilities, including SQL injection and cross-site scripting (XSS). |
| Handling Image Processing Delays | A queuing system was implemented for managing image generation requests. This allowed for efficient allocation of processing resources and helped maintain application responsiveness during high usage. Progress indicators were integrated into the user interface to inform users about the status of their image generation requests. |
| Cross-Platform Compatibility | Responsive React components were developed and optimized for various screen sizes, following a mobile-first design principle. Media queries were applied to dynamically adjust the layout, ensuring a consistent and usable experience across mobile and desktop devices. |

## Functional Testing

Functional testing was performed to verify the correct operation of each component within the DementiArt App application. This included examining the frontend and backend systems independently and their integrated functionality, specifically focusing on image generation from text prompts and the display of these images.

Frontend Testing (React): Individual React components were tested using unit testing frameworks such as Jest and React Testing Library. These tests confirmed that components rendered as expected, accurately processed user input in the text prompt field, and remained stable during updates, including the display of generated images after the "Generate" button was activated. Integration testing validated the data flow between the frontend and backend systems, particularly the transmission of the user's prompt and the reception of the generated image. Specific functionalities, such as navigation between the home screen, the prompt input screen, and the memory display screen, were tested to ensure correct operation.

Backend Testing (FastAPI): The FastAPI backend was tested to ensure that API endpoints correctly received user-provided text prompts and initiated the image generation process. Unit tests were performed on the defined routes to verify accurate data processing and the proper interaction with the Stable Diffusion model. The backend's API endpoint responsible for receiving the prompt and returning the generated image URL was thoroughly tested. Interactions with the PostgreSQL database and Cloudinary cloud image storage for saving and retrieving image data were also tested for correct execution.

## Performance and Load Testing

A key objective was to ensure DementiArt App ability to scale with increased user load and image generation requests. Performance and load testing were conducted to assess the system's capacity under high usage conditions, particularly focusing on the image generation process.

**Backend Performance (FastAPI):** Load testing tools, such as Locust, simulated a significant volume of API requests for image generation. The backend's response time under this simulated load was measured. This testing aimed to identify potential performance bottlenecks within the API requests, which informed subsequent optimizations to the FastAPI backend. These optimizations included configuration adjustments and efficient handling of requests to the Stable Diffusion model.

**Image Generation Performance (Stable Diffusion):** Given the computational demands of image generation using the Stable Diffusion model, its performance was evaluated to ensure efficient image processing, even during periods of high demand. The performance of the cloud infrastructure supporting the AI model was also monitored to confirm its ability to handle concurrent image generation requests without significant delays.

**Frontend Load:** Load testing was carried out on the frontend to ensure the React components could effectively handle many concurrent users accessing the application, particularly the loading and display of images. This testing focused on verifying the continued responsiveness of the user interface when accessed by multiple users simultaneously. Performance analysis tools, such as Lighthouse, were also utilized to measure frontend performance and identify areas for optimization, such as improving image loading times.

## Security Testing

Protecting user interactions and generated images was a primary concern in DementiArt App 's development. Security testing was performed to confirm the application's protection against common web vulnerabilities and adherence to relevant data protection standards.

**API Security:** The FastAPI backend was scanned for common web vulnerabilities, including potential injection attacks. JSON Web Tokens (JWT) were likely used for secure communication, and testing would have aimed to ensure the security of this process.

**Database Security (PostgreSQL):** Security measures for the PostgreSQL database were tested to ensure the confidentiality and integrity of stored image metadata. This included verifying appropriate access controls and secure data handling practices.

Image Storage Security (Cloudinary): As Cloudinary was used for storing user-generated images, ensuring their secure storage and delivery was essential. Testing focused on verifying that images were stored securely and accessed via secure URLs.

## User Acceptance Testing (UAT)

Following unit testing, integration, and performance testing, User Acceptance Testing (UAT) was undertaken with five participants. These participants were recruited through personal contacts and had varying levels of technical experience. The group included three individuals aged 25-40 who regularly use web-based applications and two individuals aged 50-65 with moderate experience using online tools.

During the UAT session, participants were asked to navigate to the DementiArt App homepage and enter at least three different text prompts of their choosing. They were encouraged to experiment with different types of prompts, ranging from simple nouns to more complex descriptive phrases. For each generated image, participants assessed its relevance to their prompt, its visual quality, and the speed of generation. Following their interaction with the app, participants completed a short online questionnaire that asked for feedback on the app's ease of use, the intuitiveness of the interface, and any bugs or areas for improvement they noticed.

Feedback indicated that the app was generally easy to understand and use. However, minor enhancements were suggested regarding the placement of the 'Generate' button and the clarity of the loading indicator. No critical bugs were identified. Participants tested the app on both desktop computers and mobile phones, and the feedback indicated that the app was responsive and accessible across multiple devices.

.

## Continuous Testing and Monitoring

A continuous assessment mechanism was established for DementiArt App. This involved integrating automated tests into the continuous integration (CI) pipeline to ensure that new code changes did not introduce regressions or negatively impact existing functionality, particularly related to image generation and display. Real-time monitoring tools were also implemented to track application performance metrics, monitor for errors, and identify potential issues after deployment. This monitoring system enabled the development team to respond promptly to any identified problems and ensure the stable and reliable operation of the application post-release.

## Overview of the Final System

The final version of DementiArt App is a web application designed to enable users to generate images from text prompts using AI. The application features a frontend built with React and a backend powered by FastAPI. The Stable Diffusion model is utilized for AI-based image creation. The app offers a user interface accessible through web browsers. Images are stored using Cloudinary, and user-related data is managed within a PostgreSQL database. The system's core functionality involves processing user prompts to generate images and providing a way to view previously generated images.

A colorful brain with a paintbrush and leaves

AI-generated content may be incorrect.

**Figure 7:** Logo of DementiArt App

DementiArt App offers a simple and clean interface for generating images from text prompts and viewing past creations. The application aims to provide a basic tool for creative image generation. It utilizes AI-accelerated image generation and cloud-based storage. The main functionalities and interface screens are described below

A screenshot of a sign up form

AI-generated content may be incorrect.

**Figure 8:** Opening Screen of the App

## Home Screen and Main Navigation

### The Home Screen is the initial point of interaction, presenting users with two main options:

### Create a New Memory: Navigates the user to the prompt input screen for generating an AI image.

### Play Old Memories: Displays a grid of previously generated images.

### A screenshot of a memory app AI-generated content may be incorrect.

**Figure 9:** Opening Screen of the App

### User Interface & Accessibility

The user interface is designed for ease of use, featuring large buttons. Navigation is straightforward. The calm, pastel-hued background aims to enhance usability. The React frontend and FastAPI backend ensure efficient, secure, and responsive performance.

### Prompt Input and Image Generation

Selecting "Create a New Memory" directs users to a screen with a text input field labelled "Enter your text prompt here..." and a "Generate" button.

A screenshot of a cell phone

AI-generated content may be incorrect.

**Figure 10:** Custom Prompt Input Screen

Users type their desired text prompt and click "Generate." The React frontend sends the prompt to the FastAPI backend. The backend uses the Stable Diffusion model to generate an image and stores the image URL (from Cloudinary) and potentially metadata in PostgreSQL. The backend then returns the image URL to the frontend for display.

### Viewing Old Memories

Selecting "Play Old Memories" from the home screen navigates users to a screen displaying a grid of previously generated images.

A screenshot of a phone

AI-generated content may be incorrect.

**Figure 11:** Old Memories Display Screen

This screen retrieves image metadata from the PostgreSQL database (with the actual images stored in Cloudinary) and displays them in a visual grid. Users can scroll to view past creations. The React frontend handles the rendering.

### User Interaction Flow

The user experience is direct for image generation and viewing.

Text Prompt Input: Users enter text on the "Create a New Memory" screen.

AI Image Generation: Clicking "Generate" initiates image creation via the backend and AI model.

Image Display: The generated image is shown on the screen.

Memory Retrieval: Previously generated images can be viewed on the "Play Old Memories" screen.

### Final Product Design and UI

DementiArt App features a simple and clean user interface with a pastel colour scheme. The home screen offers clear choices for image generation and viewing. The prompt input screen is direct, and image display is immediate. The "Play Old Memories" screen presents images in a grid format. The app is responsive in web browsers.

A screenshot of a phone

AI-generated content may be incorrect.

**Figure 12:** Example of a Generated Image Displayed

The interface is designed for easy navigation with clear visual elements.

### Comparison with Initial Expectations

The deployed DementiArt App provides the core functionalities of generating images from text and viewing them. Some initially described features are not present in the current live application. The fundamental aspects of prompt-based image creation and retrieval are implemented.

### 1.13 Limitations and Future Enhancements

Limitations include the absence of features like memory categorization and advanced user interaction flows. Future enhancements could involve implementing memory categorization, user accounts, and potentially editing or annotating generated memories. Further development could explore more complex AI-driven features.

# Appraisal

The DementiArt App project aimed to develop a cognitive engagement tool for individuals with dementia by integrating artificial intelligence (AI) with user-centered design. A review of the project identifies areas for potential improvement and key strengths.

Future development should focus on enhancing the AI models through improved training and fine-tuning. Increasing the accuracy and contextual relevance of the generated images in response to user prompts would improve the application's utility. Utilizing a larger, higher-quality dataset for initial model training is recommended to achieve this. Furthermore, optimizing the AI inference pipeline could lead to faster response times for image generation.

The extent of user testing during development is another area for potential improvement. While feedback from User Acceptance Testing (UAT) participants provided initial insights into usability, more frequent and iterative testing with feedback loops could further refine the application's practicality, usability, and accessibility. Additionally, incorporating multi-language support in future versions could broaden the application's reach.

Several successful aspects of the project should be continued. The selection of FastAPI as the backend framework facilitated communication between the frontend and AI models and supported efficient deployment and scalability. The implementation of large, easily clickable buttons and a high-contrast color scheme improved usability, aligning with universal design principles beneficial for individuals with cognitive impairments. The establishment of data privacy practices, including secure API communication, is necessary for responsible handling of user data.

The project demonstrated the importance of user-centered design in the development of AI-driven tools. While AI is central to the image generation, the application's design must ensure it remains adaptable and easy to use for individuals with diverse cognitive profiles. Continuous user feedback throughout the development process is crucial for creating an effective application. For similar projects, prioritizing and integrating ongoing user feedback from the initial stages, along with focusing on accessibility features, is recommended. Moreover, considering scalability, AI model updates, and system extensibility are important for long-term viability. For any application handling potentially sensitive data, especially for vulnerable groups, a robust security architecture is necessary.

Feedback from the User Acceptance Testing indicated that DementiArt App achieved its basic objective of providing cognitive stimulation through AI-generated images based on text prompts. Future development could benefit from incorporating mechanisms to collect user feedback on the generated images to improve their relevance and quality. Addressing the identified areas will contribute to a more effective and scalable application.

# Summary and Conclusions

The DementiArt App project aimed to develop a web application to provide cognitive stimulation for individuals with dementia through AI-generated images based on text prompts. The application utilized AI for image generation and potentially for enhancing the generated content. The project goal was to create a user-friendly platform with features designed to maintain user confidentiality and data security.

The application employed a client-server architecture with a FastAPI backend, Stable Diffusion for AI image generation, and a React frontend. Cloud storage was used for image management and access. The development process included unit tests for backend components, integration tests for API interactions, and user tests with individuals possessing varying technical experience to evaluate usability. Performance testing was also conducted to assess application responsiveness.

Key outcomes indicated the feasibility of using AI in dementia care for engaging individuals with AI-generated images from text inputs. User feedback from testing suggested the application was generally easy to understand and use, with minor recommendations for interface adjustments. The user interface featured large buttons and a clear layout, intended to improve accessibility for users with cognitive impairments. Data handling procedures aimed to ensure user privacy. Challenges encountered included the need for further refinement in the relevance and consistency of AI-generated content. These were addressed through ongoing adjustments during development. User feedback informed the final version of the application, contributing to improvements in its technical aspects and practicality for users.

In conclusion, DementiArt App achieved its primary objective of providing a tool for cognitive stimulation through AI-generated images from text prompts. The project demonstrated the potential of technology to support individuals experiencing memory loss. While the current application is functional, future development could focus on areas such as improving scalability and enhancing the AI model's performance. This project provided valuable insights for future development in AI-based tools for dementia care.

# Future Work

The current version of DementiArt App demonstrates the potential of AI to support dementia care by utilizing AI-generated images from structured text input to provide a potentially therapeutic and engaging experience for users. However, several key areas require further development to enhance the system’s usability, scalability, and effectiveness.

One primary area for future work is the integration of multi-language support. Currently, the system is only available in English, limiting its utility. Implementing multilingual support is necessary to broaden the application's accessibility. This would require training the AI models with datasets in different languages to ensure the generated content retains its coherence and relevance across various cultural contexts.

Another aspect for future development is the incorporation of real-time AI learning and personalization. The current system generates static images. Future iterations could implement adaptive learning based on user behavior, user feedback, and preferences. This would enable the system to provide more personalized and relevant content, potentially leading to more effective therapeutic outcomes and a stronger connection to memories with continued use.

Furthermore, reducing response times and increasing computational performance are essential. While initial efforts have been made to decrease latency, image generation can still take time. Exploring lighter AI models, edge computing approaches, or hybrid cloud architectures could enable faster and smoother interactions.

The system’s usability could also be improved through expanded user testing. While input from caregivers and initial user testing has been valuable, broadening the testing to include a wider range of participants, especially individuals at various stages of dementia, would yield further insights. Additionally, clinician-led assessments could help determine the system's therapeutic efficacy and guide its integration into dementia care pathways as a validated tool.

Developing a mobile application version of DementiArt App could enhance accessibility and user engagement. A dedicated iOS and Android application could offer offline functionality, custom features tailored to mobile use, and increased convenience for both users and caregivers. This would also improve the system's portability and adaptability for use in various settings.

Finally, ensuring robust long-term data privacy and security is crucial. While the current technology utilizes transient storage and secure communication, future updates could incorporate advanced security measures and explore privacy-preserving AI techniques to enhance data protection. Adherence to global data protection regulations, such as GDPR, is essential to build trust and ensure compliance.

In general, future development of DementiArt App should aim for enhanced accessibility, a personalized AI-driven experience, optimized computational efficiency, and robust data privacy measures. These enhancements will contribute to making the system a more effective tool for dementia care, benefiting both individuals with dementia and their caregivers. Further innovation has the potential to establish DementiArt App as a significant model for AI-backed dementia care, positively impacting the cognitive wellness and quality of life of people with dementia.

# Futuristic Idea

**Sensory Resonance Stimulation for Memory Activation**

The core principle of this concept is to leverage the strong connection between sensory experiences and memory formation. Unlike methods relying solely on visual or auditory cues, Sensory Resonance Stimulation aims to directly activate brain regions associated with specific memories by recreating the sensory context of those experiences.

**Olfactory Stimulation: Evoking the Past Through Scent**

**Mechanism:** Olfactory pathways have a unique and direct connection to the limbic system, which plays a crucial role in emotion and memory. This direct link makes scents powerful triggers for recalling deeply ingrained memories.

**Micro-Aerosol Delivery System:** A miniaturized, wearable device (potentially integrated into a Bio-Integrated Memory Resonance System headset) could house a library of synthesized scent molecules. This system would be capable of releasing precise and subtle bursts of specific scents.

**Personalized Scent Library:** Through initial data gathering, potentially involving interviews with the patient and family, as well as analysis of personal items, a personalized scent library could be created. This library would contain scents associated with significant life events, cherished moments, and familiar environments. Examples include:

The aroma of a specific type of flower from a childhood garden.

The scent of a particular dish cooked by a loved one.

The smell of woodsmoke from a memorable camping trip.

The unique fragrance of a specific brand of soap used in their younger years.

A person smoking a cigarette

AI-generated content may be incorrect.

Figure 13: Example of a Generated Image Displayed

**Olfactory Stimulation: Evoking the Past Through Scent**

**Mechanism:** Olfactory pathways are uniquely and directly connected to the limbic system, which plays a crucial role in emotion and memory. This direct link makes scents powerful triggers for recalling deeply ingrained memories.

**Micro-Aerosol Delivery System:** A miniaturized, wearable device (perhaps integrated into the BIMRS headset) **Bio-Integrated Memory Resonance System** would house a library of synthesized scent molecules. This system would be capable of releasing extremely precise and subtle bursts of specific scents.

**Personalized Scent Library:** Through initial data gathering (potentially involving interviews with the patient and family, as well as analysis of personal items), a personalized scent library would be created. This library would contain scents associated with significant life events, cherished moments, and familiar environments. Examples include:

* + - The aroma of a specific type of flower from a childhood garden.
    - The scent of a particular dish cooked by a loved one.
    - The smell of woodsmoke from a memorable camping trip.
    - The unique fragrance of a specific brand of soap used in their younger years.

# References

* Stearn, E. (2023). Is your memory loss usual for your age... or is it a sign of DEMENTIA? Experts share 5 ways to tell them apart. Daily Mail. Retrieved from https://www.dailymail.co.uk/health/article-11990201/Is-memory-loss-usual-age-sign-DEMENTIA-Experts-share-5-ways-tell-apart.html [Accessed 31 March 2025].
* Ali, Y., Caballero, G.E., Shatnawi, E., Dadich, A., Steiner-Lim, G.Z. and Karamacoska, D. (2024). Assessing the impact of an online dementia awareness initiative co-created with and for English, Arabic and Vietnamese speaking communities: a case study. Health Expectations. Available at: https://doi.org/10.1111/hex.14026 [Accessed 31 March 2025].
* Baker, F.A., MacDonald, R.A.R. and Tamplin, J. (2017). The role of music in dementia care: Exploring the artist’s perspective. Arts & Health, 9(2), pp. 144-158.
* Chatterjee, H.J. and Noble, G. (2019). The role of the arts in dementia care: Recommendations from the international research symposium on the arts in dementia care. Perspectives in Public Health, 139(4), pp. 195-197.
* Critten, V. and Kucirkova, N. (2019). Co-designing technology for dementia care. Dementia, 18(6), pp. 2305-2317.
* Fleming, S.M. et al. (2018). Art in dementia care: Benefits, challenges, and strategies for implementation. Dementia, 17(5), pp. 659-679.
* Friedberg, A., Pasquini, L., Diggs, R., et al. (2023). Prevalence, Timing, and Network Localization of Emergent Visual Creativity in Frontotemporal Dementia. JAMA Neurology, 80(4), pp. 378-386.
* Huggins, L.K.L., Min, S.H., Dennis, C.A., Østbye, T., Johnson, K.S. and Xu, H. (2022). Interventions to promote dementia knowledge among racial/ethnic minority groups: a systematic review. Journal of the American Geriatrics Society, 70(2), pp. 609–621.
* Kim, C. and Kwon, J. (2020). Exploring the relationship between creative engagement and quality of life in dementia patients: A pilot study. Journal of Applied Gerontology, 39(4), pp. 364-372.
* Lazar, A., Kucirkova, N. and Evans, L. (2018). The role of technology in supporting the arts for dementia patients. Arts & Health, 10(2), pp. 129-143.
* Liu, Y., He, M. and Wei, Y. (2021). Developing AI-driven creative arts tools for dementia patients: Feature and demand analysis. Computers in Human Behavior, 115, 106570.
* MacRitchie, J., Floridou, G.A. and Christensen, J. (2022). The use of technology for arts-based activities in older adults living with mild cognitive impairment or dementia: A scoping review. Dementia, 22(1), pp. 252-280.
* Manca, M., Fortunato, G. and Pani, F. (2021). Designing technological art tools for people with dementia. International Journal of Human-Computer Studies, 147, 102574.
* Tsekleves, E. (2021). Technology and the arts for people with dementia: Opportunities and challenges. Ageing & Society, 41(2), pp. 393-411.
* Zhou, Y., Liu, Y. and Zhang, Y. (2023). Personalized Art Therapy with AI: Enhancing Engagement among Dementia Patients. Journal of Medical Internet Research, 25, e43200.
* Alexander, P. (2024). I got dementia at 49 - here's the first sign my brain was in decline. Daily Mail. Retrieved from https://www.msn.com/en-gb/health/health-news/i-got-dementia-at-49-here-s-the-first-sign-my-brain-was-in-decline/ar-AA1y3zGt?ocid=msedgntp&pc=HCTS&cvid=475a0e63980d4f1c895355f355f9324b&ei=55 [Accessed 31 March 2025].
* Lulle, K., Agrawal, P., Amrutwar, M., & Khiani, S. (2024, December). An AI-Powered Application to Enhance Cognitive Health and Provide Caregiver Support for Dementia Patients. Paper presented at the 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), IEEE, pp. 93-98.
* Kanchanamala, P., Chippada, R., Bennabatthula, T., & Muppidi, S. (2024, September). AI-Powered Alzheimer's Care: Smart Face Recognition, Medicine Reminders, and Location. Paper presented at the 2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon), IEEE, pp. 1-8.
* Bint Khalid, U., Naeem, M., Stasolla, F., Syed, M. H., Abbas, M., & Coronato, A. (2024). Impact of ai-powered solutions in rehabilitation process: Recent improvements and future trends. International Journal of General Medicine, 17, 943.
* Khalid, U. B., Naeem, M., Stasolla, F., Syed, M. H., Abbas, M., & Coronato, A. (2024). Impact of AI-powered solutions in rehabilitation process: Recent improvements and future trends. International Journal of General Medicine, pp. 943-969.
* Milella, F., Russo, D., & Bandini, S. (2023). AI-powered solutions to support informal caregivers in their decision-making: a systematic review of the literature. OBM GERIATRICS, 7(4).
* Velagaleti, S. B. (2023, December). A Study on Feasibility and Acceptability of an AI-Powered VR/AR Cognitive Rehabilitation Platform for Patients with Alzheimer’s Disease and Dementia. Paper presented at the International Conference on Data Science, Machine Learning and Applications, Springer Nature Singapore, pp. 1079-1083.
* Andargoli, A. E., Ulapane, N., Nguyen, T. A., Shuakat, N., Zelcer, J., & Wickramasinghe, N. (2024). Intelligent decision support systems for dementia care: A scoping review. Artificial Intelligence in Medicine, 102815.
* Tomar, V., Kumar, A., Kate, V., Kumar, S., & Gottumukkala, P. (2024). Advancements in Alzheimer's Diagnosis: A Comprehensive Exploration of AI-Powered Diagnostic Tools and Software. In: AI-Driven Alzheimer's Disease Detection and Prediction, IGI Global, pp. 69-80.
* Parsapoor, M., Alam, M. R., & Mihailidis, A. (2022). AI-powered language assessment tools for dementia. arXiv preprint arXiv:2209.12652.
* Tariq, M. U. (2025). AI-Powered Breakthroughs: Revolutionizing Cognitive Psychology and Neuropsychology With Machine Learning. In: Transforming Neuropsychology and Cognitive Psychology With AI and Machine Learning, IGI Global Scientific Publishing, pp. 65-92.
* Pr, V., Dharsan, S., & Aravindan, M. S. (2023, November). Dementia Care Using AI: Real-time Patient Trajectory Monitoring System. Paper presented at the 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), IEEE, pp. 1-6.
* Parsa, M., Alam, M. R., & Mihailidis, A. (2021). Towards AI-powered language assessment tools.
* Hasan, W. U., Zaman, K. T., Wang, X., Li, J., Xie, B., & Tao, C. (2024). Empowering Alzheimer’s caregivers with conversational AI: A novel approach for enhanced communication and personalized support. npj Biomedical Innovations, 1(1), 1-10.
* KS, A. K., Gireesh, H. R., & Shashidhar, V. (2024). Revolutionizing Alzheimer's Diagnosis: Navigating the Challenges and Embracing Opportunities in the Clinical Integration of AI-Powered Tools. In: AI-Driven Alzheimer's Disease Detection and Prediction, IGI Global, pp. 160-174.
* Sugumar, D., Sujatha, S., & Veeramakali, T. (2024). Patient-Centered AI Solutions for Managing Alzheimer's Disease. In: AI-Driven Alzheimer's Disease Detection and Prediction, IGI Global, pp. 250-263.
* Kale, M. B., Wankhede, N. L., Pawar, R. S., Ballal, S., Kumawat, R., Goswami, M., ... & Koppula, S. (2024). AI-driven innovations in Alzheimer's disease: Integrating early diagnosis, personalized treatment, and prognostic modelling. Ageing Research Reviews, 102497.
* Joshi, M. (2024). An AI-Powered Smart Watch Application for Personalized, Adaptive, Music-Based Therapy to Manage Anxiety in Patients With Alzheimer’s (Master's thesis). New York University Tandon School of Engineering.
* Muppavaram, K., Gangopadhyay, A., Ramadass, S., & Prakash, N. (2024). Real-World Impact: Case Studies and Success Stories in AI-Driven Alzheimer's Disease Research and Care. In: AI-Driven Alzheimer's Disease Detection and Prediction, IGI Global, pp. 237-249.
* Sun, F., Jiang, L., Chen, X. S., & Feng, Y. (2025). Interactive AI Technology for Dementia Caregivers: Needs and Implementation Evidence. Journal of Technology in Human Services, 1-26.
* Kim, S., Han, D. Y., & Bae, J. (2024). Transforming Alzheimer’s Digital Caregiving through Large Language Models. Current Alzheimer Research, 21(7), 503-516.
* Gonçalves, M. A. C. (2024). AI for Alzheimer: Can Artificial Intelligence Have a Positive Impact on Alzheimer's Care? Understand the Potential Benefits of AI (Master's thesis). Universidade NOVA de Lisboa (Portugal).
* Khan, Z., Sami, M., & Bhatti, S. An Artificial Intelligence based Approach to Assist Stage 1 to 4 (from no impairment to moderately declined) Alzheimer’s Disease Patients and Caregivers.
* Ali, H. (2022). AI in neurodegenerative disease research: Early detection, cognitive decline prediction, and brain imaging biomarker identification. Int J Eng Technol Res Manag, 6(10), 71.
* Gupta, A., Sawhney, S., & Ahmed, S. (2025). CareTaker. ai—A Smart Health-Monitoring and Caretaker-Assistant System for Elder Healthcare. Engineering Proceedings, 78(1), 7.
* Xygkou, A., Ang, C. S., Siriaraya, P., Kopecki, J. P., Covaci, A., Kanjo, E., & She, W. J. (2024, May). MindTalker: Navigating the Complexities of AI-Enhanced Social Engagement for People with Early-Stage Dementia. Paper presented at the 2024 CHI Conference on Human Factors in Computing Systems, pp. 1-15.
* Aburub, F., & Agha, A. S. A. (2024, February). AI-driven psychological support and cognitive rehabilitation strategies in post-cancer care. Paper presented at the 2024 2nd International Conference on Cyber Resilience (ICCR), IEEE, pp. 1-6.
* Malik, J. B. (2023). Artificial Intelligence Enabled Healthcare: Opportunities and challenges for patients with dementia.
* Krueger, J. (2025). Home as mind: AI extenders and affective ecologies in dementia care. Synthese, 205(2), 1-22.
* Jain, R. (2024). NeuroDolphin–An AI-Enabled Mobile Application for Alzheimer’s Dementia (Master's thesis). Northeastern University.
* Kontos, P., Dupuis, S., Colobong, R., Delaney, E., Gertin, R., Engell, K., & MacTavish, E. (2024). EXPLORING END-OF-LIFE DEMENTIA CARE IN RESIDENTIAL CARE SETTINGS: TOWARD A RELATIONAL MODEL. Innovation in Aging, 8(Supplement\_1), 732-732.4.
* Gadde, H. (2024). AI-Powered Fault Detection and Recovery in High-Availability Databases. International Journal of Machine Learning Research in Cybersecurity and Artificial Intelligence, 15(1), 500-529.
* Giansanti, D., & Pirrera, A. (2025). Integrating AI and Assistive Technologies in Healthcare: Insights from a Narrative Review of Reviews. Healthcare, 13(5), 556.
* Burri, S. R., Ghorpade, V. V., Dutt, V., & Lipi, K. (2023, November). The Rise of Virtual Health Assistants: Chatbot-Based Healthcare Support and Counseling Using Recurrent Neural Networks (RNNs). Paper presented at the 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), IEEE, pp. 811-816.
* Manoharan, G., Pant, L. M., Tewari, R., Vichare, S., Sarkar, P., & Dhanraj, J. A. (2024, September). AI-Powered Chatbots for Mental Health Support. Paper presented at the 2024 7th International Conference on Contemporary Computing and Informatics (IC3I), IEEE, Vol. 7, pp. 436-440.
* Nebeker, C., Parrish, E. M., & Graham, S. (2022). The AI-powered digital health sector: ethical and regulatory considerations when developing digital mental health tools for the older adult demographic. In: Artificial Intelligence in Brain and Mental Health: Philosophical, Ethical & Policy Issues, Cham: Springer International Publishing, pp. 159-176.
* Karami, V., Yaffe, M. J., Gore, G., Moon, A., & Rahimi, S. A. (2024). Socially Assistive Robots for Individuals with Alzheimer's Disease: A Scoping Review. Archives of gerontology and geriatrics, 105409.
* Khang, A. (Ed.). (2024). Driving Smart Medical Diagnosis Through AI-Powered Technologies and Applications. IGI Global.
* Arango, J. A. R., Marco-Detchart, C., & Inglada, V. J. J. (2025). Personalized Cognitive Support via Social Robots. Sensors (Basel, Switzerland), 25(3), 888.
* Langston, E. M., Hattakitjamroen, V., Hernandez, M., Lee, H. S., Mason, H. Ç., Louis-Charles, W., ... & Boot, W. R. (2025). Exploring Artificial Intelligence-Powered Virtual Assistants to Understand their Potential to Support Older Adults’ Search Needs. Human Factors in Healthcare, 100092.

**Appendix: Figures**

* Figure 1: Possible Signs of Dementia (Stearn, 2023)
* Figure 2: Growth in Estimated Cases of Dementia (2017-2040) (Stearn, 2023)
* Figure 3: Thematic Map
* Figure 4: Deployment Flow of Dementi App
* Figure 5: Code for Prompt Construction and API Interaction
* Figure 6: Database Connection Configuration with SQLAlchemy

**Appendix: Download App**

**https://memory-generator-app.netlify.app/**

<Signup with Gmail Account>

A qr code with blue dots

AI-generated content may be incorrect.