Solution Architecture Summary: Alzheimer Detection System

1. Introduction:

The Alzheimer Detection System combines frontend technologies, backend processing, and machine learning for early Alzheimer's detection. This report outlines the architecture facilitating user interaction, model prediction, and result visualization.

2. Frontend:

2.1 UI:

Bootstrap and EJS create a user-friendly interface for image uploads.

2.2 NodeJS and Express:

NodeJS with Express.js manages user requests, rendering views, and communication with the backend.

3. Backend:

3.1 Flask API:

The Flask API processes image data, invoking a CNN model with SMOTE for improved predictions.

3.2 Model:

The CNN model, saved in H5 format, efficiently predicts Alzheimer's likelihood.

4. Integration:

4.1 User Interaction:

Users upload images, and NodeJS forwards data to the Flask API.

4.2 Processing:

The Flask API employs the model, and results are sent to the frontend.

4.3 Visualization:

ChartJS dynamically displays bar charts representing prediction probabilities.

5. Deployment:

5.1 Cloud Hosting:

Deployable on cloud platforms for scalability.

5.2 Containerization:

Docker containers encapsulate components for easy deployment.

6. Scalability and Extensibility:

Designed for scalability with cloud services and containerization. Extensible for future enhancements.
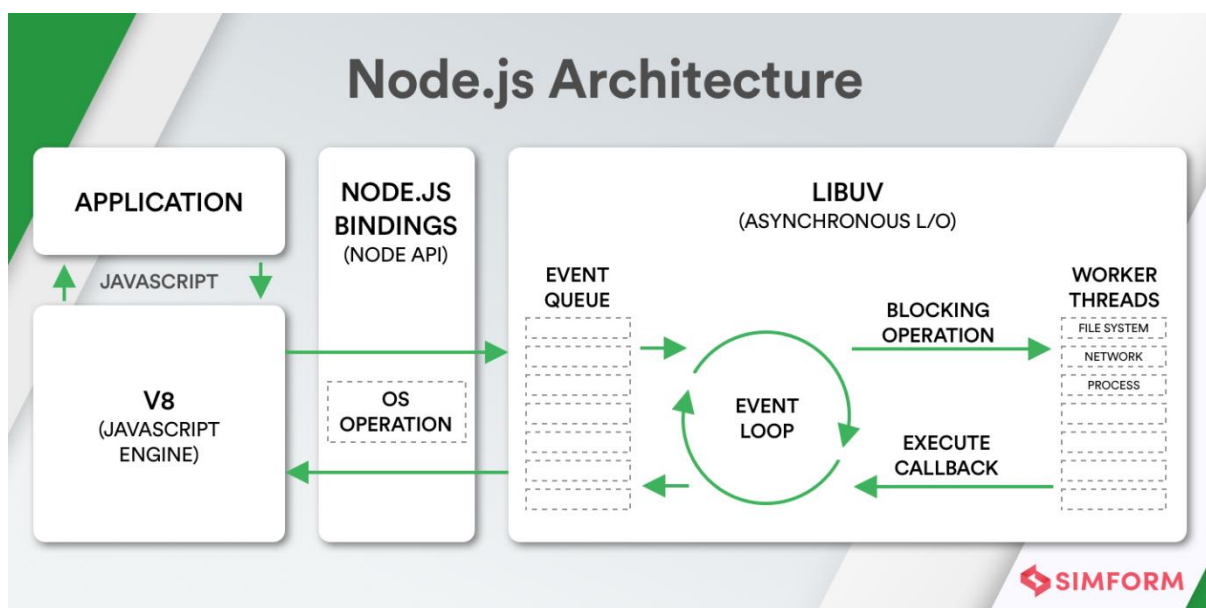
7. Security:

Implement secure communication with HTTPS and access controls for data protection.

8. Conclusion:

The Alzheimer Detection System's modular architecture integrates frontend, backend, and machine learning, providing a flexible, scalable, and accurate tool for Alzheimer's prediction

Architecture:

DataFlow Diagram:

D User

JS Frontend ← **flaskAPI** get predictions → **D** model.h5

fetch predictions from flask API run the model

render html templates for frontend

nodeJS server