

Project Design Phase-II Technology Stack (Architecture & Stack)

Date	
Team ID	
Project Name	Depression: A common mental health Disorder
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

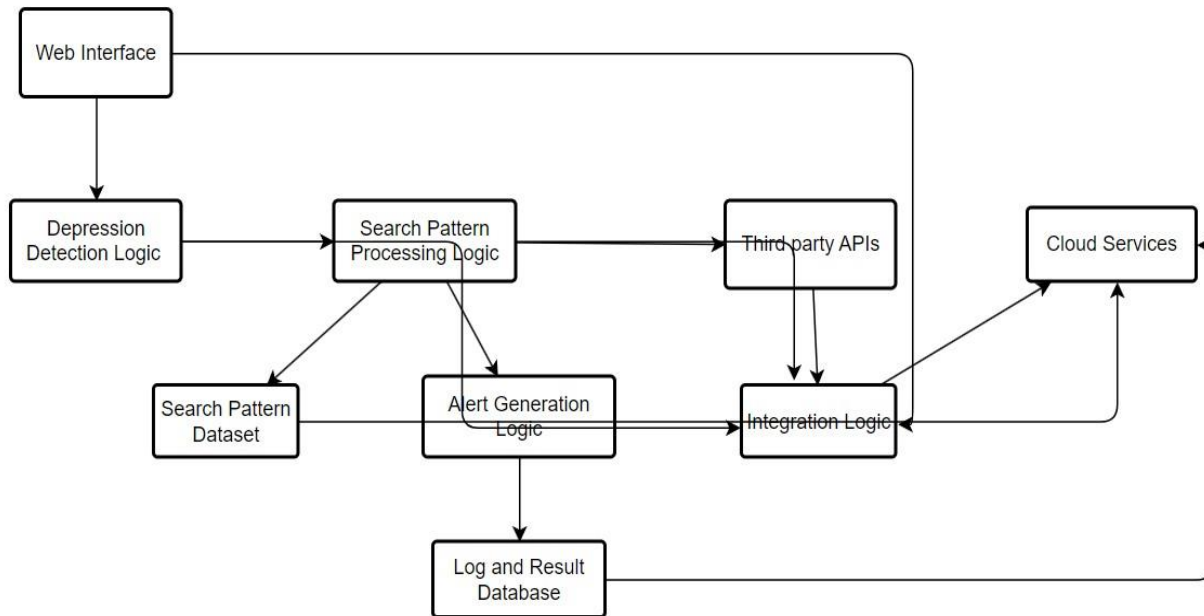


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Presents analysis results to the user in a clear and sensitive manner.	React Native for cross-platform mobile app development.
2.	Application Logic-1	Responsible for collecting and processing search history data on the user's device.	Android's Kotlin for Android version or Swift for iOS version.
3.	Database	Stores non-sensitive user data like user profiles, preferences, and basic search history.	PostgreSQL for relational database management
4.	File Storage/ Data	Stores multimedia content, if any, related to the user's mental health journey	Amazon S3 for scalable and secure object storage.
5.	Frame Work	Used to develop the plugin and integrate frontend and backend.	
6.	Deep Learning Model	Analyzes search patterns to identify potential signs of depression.	TensorFlow or PyTorch for creating and training machine learning models.
7.	Infrastructure (Server / Cloud)	Hosts the backend logic, databases, and handles server-side operations.	Amazon Web Services (AWS) for scalability, reliability, and cloud services.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source frameworks to encourage community contributions and transparency.	React Native for the user interface, Django for backend (open-source frameworks).
2.	Security Implementations	Implements end-to-end encryption, secure data transmission, and follows industry-standard security	HTTPS for secure communication, Firebase

		protocols.	Authentication for user identity verification
3.	Scalable Architecture	Designed with a modular structure to easily scale with increased user data and feature enhancements.	Microservices architecture using Docker and Kubernetes for containerization and orchestration
4.	Availability	Ensures 24/7 availability through load balancing and redundancy in server deployment	Content Delivery Network (CDN) for efficient content distribution, AWS Auto Scaling for automatic resource adjustments
5.	Perfromance	Optimized code and database queries for responsive and efficient user experience	Caching mechanisms using Redis for quick data retrieval, performance monitoring tools like New Relic

References:

<https://reactnative.dev/docs/getting-started>

<https://docs.djangoproject.com/en/5.0/>

https://www.tensorflow.org/api_docs

<https://docs.aws.amazon.com/>

<https://owasp.org/site-documentation/>

<https://kubernetes.io/docs/home/>