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

Date	November 16, 2023
Team ID	592207
Project Name	Predicting Mental Health Illness of Working Professionals Using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:

The mental health prediction system utilizes a combination of user-friendly technologies, robust security measures, and scalable architecture to provide accurate and reliable predictions. The user interface is built using HTML, CSS, and JavaScript, ensuring a seamless and interactive experience for users. Python serves as the primary programming language for application logic and machine learning model development, leveraging its versatility and extensive ML libraries. Data storage is handled by relational databases like PostgreSQL or MySQL, while cloud-based databases from AWS RDS or Azure Cosmos DB provide flexibility and scalability. External APIs facilitate data exchange and integration with third-party services, while cloud-based storage solutions like AWS S3 or Google Cloud Storage manage file storage requirements.

The system's open-source frameworks include Django and Flask, offering flexibility and ease of development. Security is paramount, with HTTPS encryption protocols and JWT-based authentication protecting user data. Microservices architecture ensures scalability, enabling the system to handle increasing demands effectively. Load balancers and distributed server architecture ensure availability, minimizing downtime and maintaining responsiveness. Caching mechanisms and CDNs optimize performance, handling a high volume of requests efficiently. Overall, the mental health prediction system showcases a well-structured and secure implementation, utilizing advanced technologies to deliver accurate predictions and support users' mental well-being

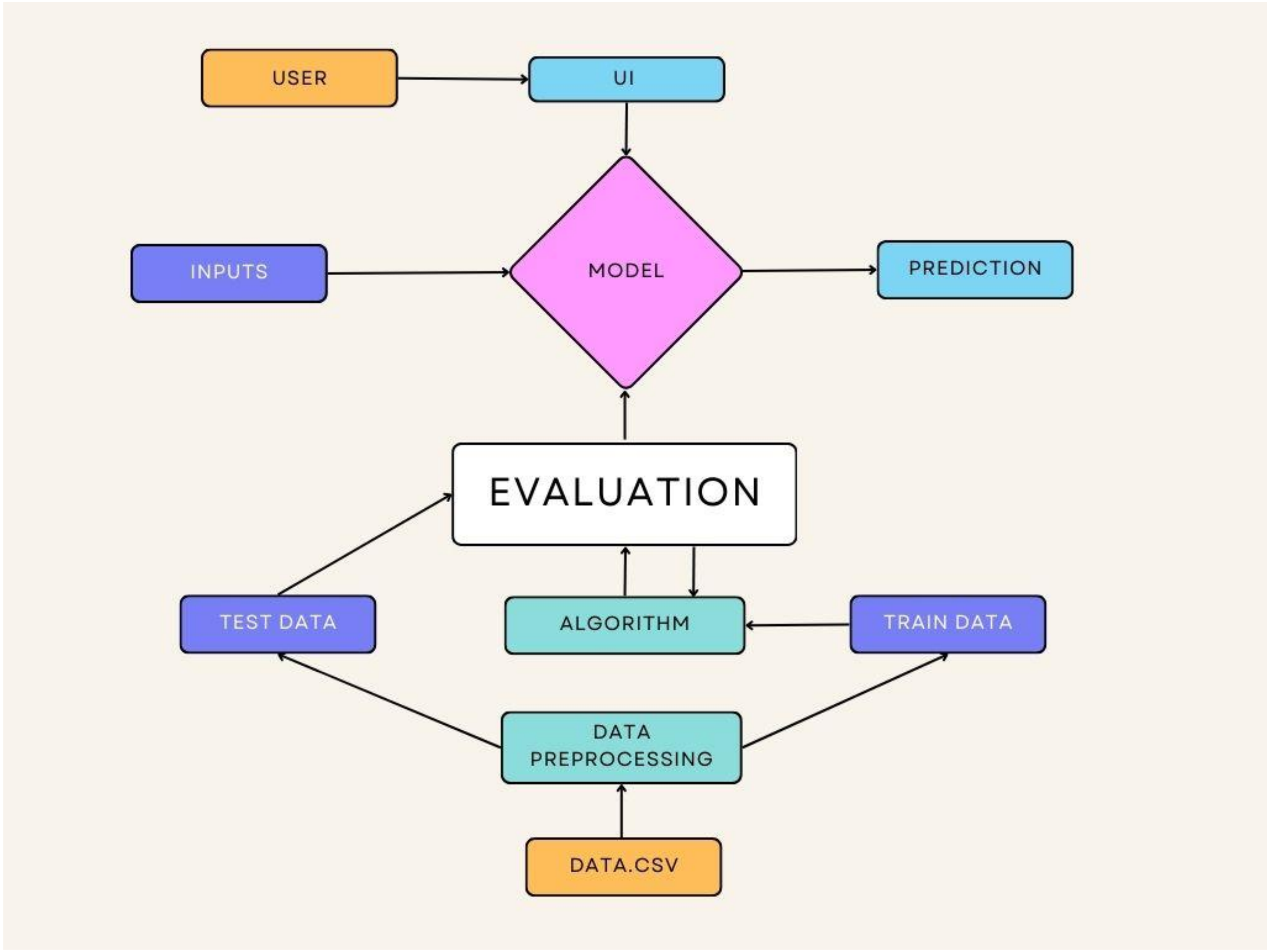
USER
UI
INPUTS
MODEL
PREDICTION
EVALUATION
TEST DATA
ALGORITHM
TRAIN DATA
DATA PREPROCESSING
DATA.CSV

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
5.	Database	Data Type, Configurations etc.	Relational Database Management System (., PostgreSQL, MySQL)
6.	Cloud Database	Database Service on Cloud	Cloud-based database service (., AWS RDS, Azure Cosmos DB)
7.	File Storage	File storage requirements	Cloud-based storage solution (., AWS S3, Google Cloud Storage)
8.	External API-1	Purpose of External API used in the application	RESTful API with JSON
10.	Machine Learning Model	Purpose of Machine Learning Model	Python-based machine learning frameworks (] TensorFlow, PyTorch)
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local Server Configuration: Web servers (e.g., Apache, Nginx)9 Cloud Server Configuration: Cloud service providers (e.g., AWS EC2, Azure VM)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Django, Flask (Python web frameworks)
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption protocols (e.g., HTTPS), JWT for authentication
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Microservices architecture

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of application (. use of load balancers, distributed servers etc.)	Load balancers, distributed server architecture
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	Caching mechanisms, Content Delivery Networks (CDNs)