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

Date	**15 October 2022**
Team ID	**PNT2022TMID46353**
Project Name	**Project - "EXPLORATORY ANALYSIS OF RAIN FALL DATA IN INDIA FOR AGRICULTURE"**
Student Name	**Akshaya.V, Priyadharshini.J, Santhini Devi.S, Swathika.G**
Student Roll No	**820319104003, 820319104029, 820319104036, 820319104044**
Maximum Marks	**4 Marks**

Technical Architecture:

Guidelines Given:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

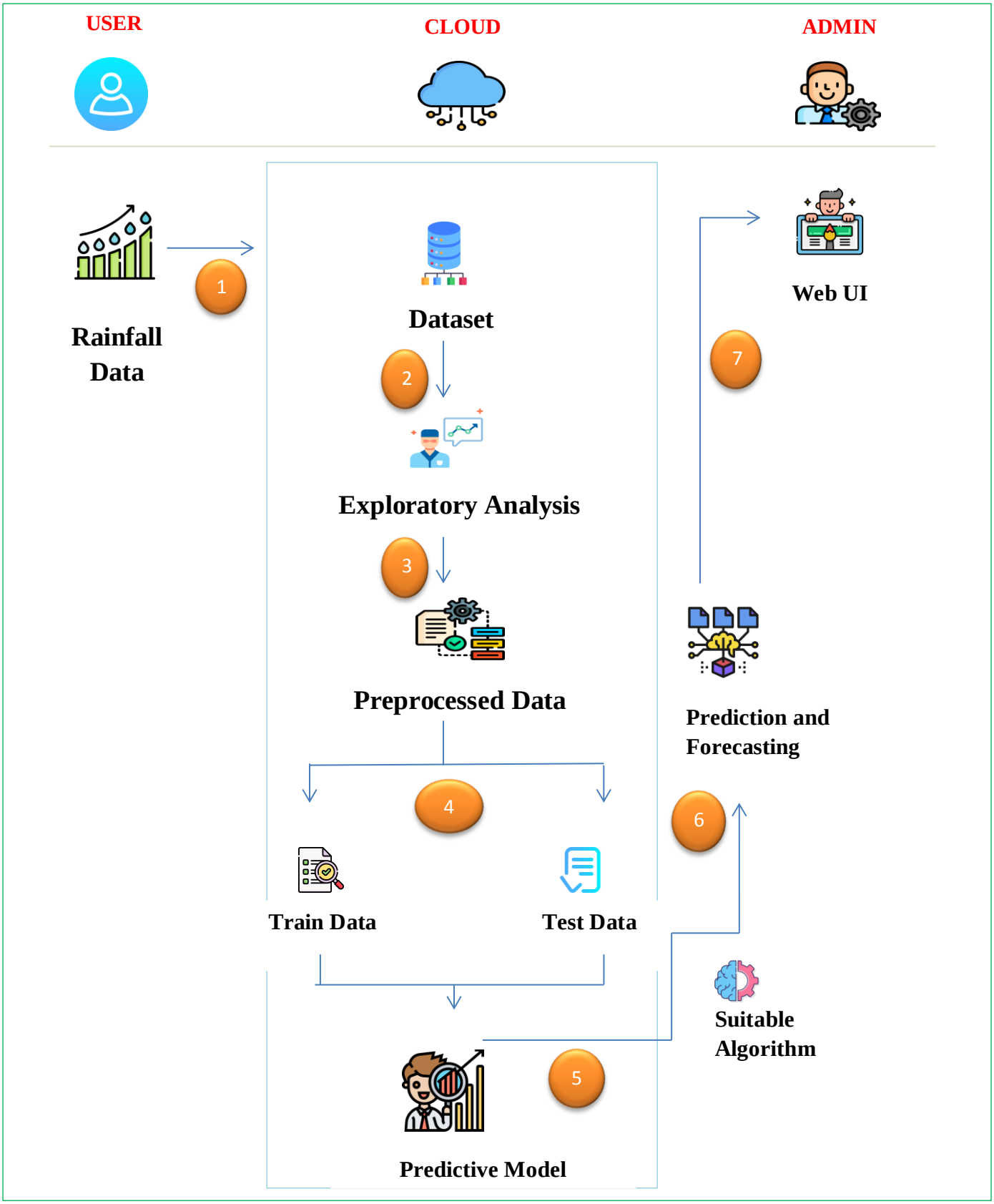
USER
CLOUD
ADMIN
Rainfall Data
1
Dataset
2
Exploratory Analysis
3
Preprocessed Data
4
Train Data
Test Data
5
Predictive Model
6
Suitable Algorithm
7
Prediction and Forecasting
Web UI

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	**User Interface**	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, Python Flask
2.	**Application Logic-1** **Login**	Logic for a process in the application	Python
3.	**Application Logic-2**	Logic for a process in the application	Python
4.	**Application Logic-3**	Logic for a process in the application	Python Flask
5.	**Database**	Data Type, Configurations etc.	MySQL
6.	**Cloud Database**	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	**File Storage**	File storage requirements	Local Filesystem
8.	**External API-1**	Purpose of External API used in the application	IBM Weather API
9.	**Machine Learning Model**	Purpose of Machine Learning Model	Predictive Modelling
10.	**Infrastructure (Server / Cloud)**	Application Deployment on Local System / Cloud Local Server Configuration: Built on Flask Web Server	Local Server

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	**Open-Source Frameworks**	List the open-source frameworks used	Micro-web Framework using Python
2.	**Security Implementations**	List all the security / access controls implemented, use of firewalls etc.	Flask Security
3.	**Scalable Architecture**	Justify the scalability of architecture (3 – tier, Micro-services)	Three-Tier Architecture
4.	**Availability**	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Load Balancers
5.	**Performance**	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	High Performance by Load Balancers