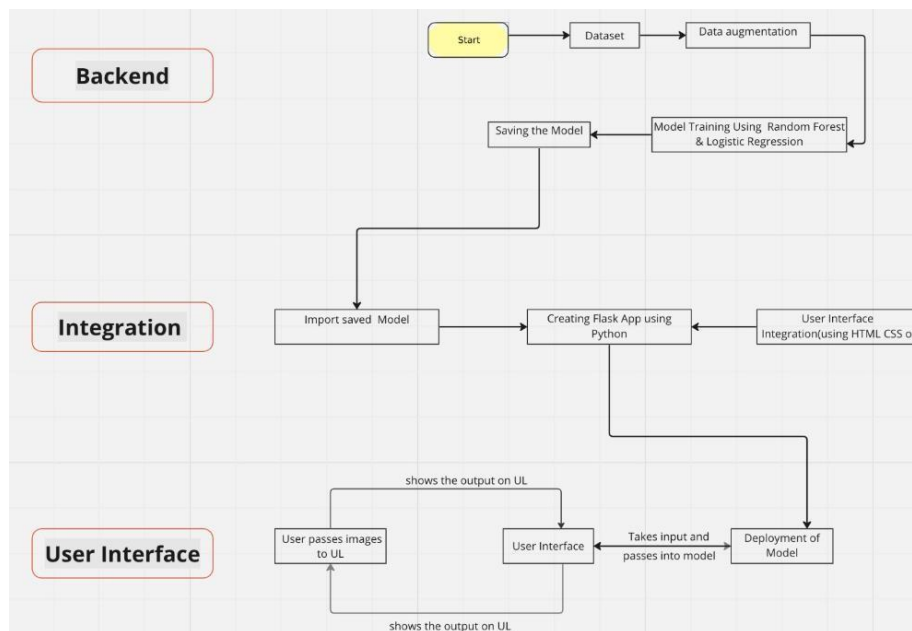


## Project Design Phase-II Technology Stack (Architecture & Stack)

Date	10 NOVEMBER 2023
Team ID	Team-592253
Project Name	FetalAI: USING MACHINE LEARNING TO PREDICT AND MONITOR FETAL HEALTH
Maximum Marks	4 Marks

### Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table2



#### Guidelines:

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)

**Table-1: Components & Technologies:**

S. No	Component	Description	Technology
1.	User Interface	The Web User Interface (WebUI) provides network administrators with a single solution for provisioning, monitoring, and optimizing devices.	HTML ,C SS , JAVASCRIPT
2.	Application Logic-1	When the user clicks the submit button the WebUI returns the prediction	JAVASCRIPT
3.	Application Logic-2	Logic for a process in the application	-
4.	Database	Data Type, Configurations etc.	-
5.	Cloud Database	Database Service on Cloud	-
6.	File Storage	File storage requirements	-
7.	External API-1	Purpose of External API used in the application	Flask routes
8.	External API-2	Purpose of External API used in the application	Flask routes
9.	Machine Learning Model	Purpose of Machine Learning Model	Predicts Output using XG BOOST ALGORITHM
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Local

**Table-2: Application Characteristics:**

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

**References:**

<https://c4model.com/> <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>  
<https://www.ibm.com/cloud/architecture> <https://aws.amazon.com/architecture> <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>