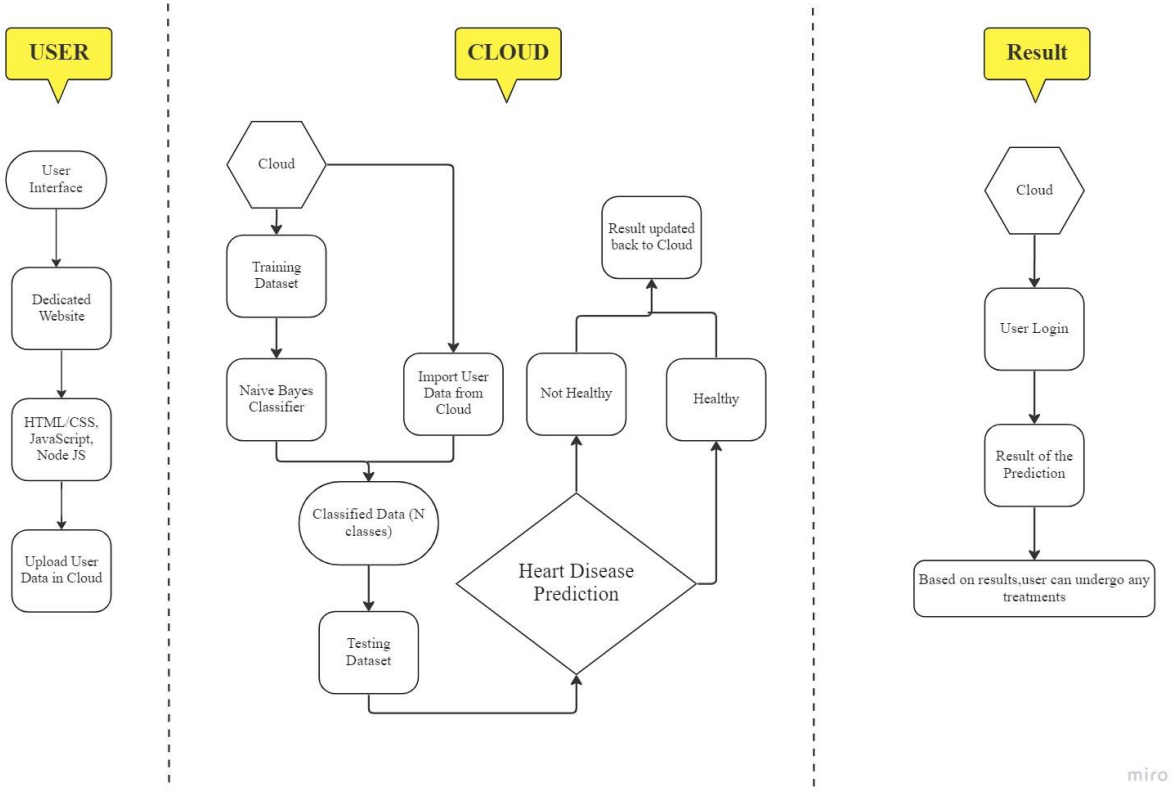
**Project Design Phase-II**
**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 03October 2022 |
| Team ID | PNT2022TMID14797 |
| Project Name | Visualizing and Predicting Heart Diseases with an Interactive Dash Board |
| Maximum Marks | 4 Marks |

**Technical Architecture:**




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 way of interaction between user and the application | HTML, CSS, JavaScript , NodeJS |
| 2. | Cloud | User data are stored in Cloud, it is storage area for the data | IBM Cloud |
| 3. | Data cleaning and pre-processing | Data cleaning is a process by which inaccurate, poorly formatted, or otherwise messy data is organized and corrected it. Data pre-processing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure | Cognos Analytic Tool |
| 4. | Training Dataset | Training data is the subset of original data that is used to train the machine learning model. | Python |
| 5. | Naive Bayes classifier | Algorithm used for the prediction of the disease | Python |
| 6. | Testing dataset | Test data is data which has been used to check the accuracy of the ML model. Testing is done with classified user data based on Naive Bayes classifier | Python |
| 7. | Result of the application | The result is based on the decision from the testing data and gives output as healthy or unhealthy | Python |
| 8. | Result to the user | The result obtained from the testing data is uploaded again back to the cloud | IBM Cloud |
| 9. | User result access | The user can view his/her result in their cloud login | IBM Cloud |

**Table-2: Application Characteristics:**

| S.No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Frameworks are about more than just creating a development environment. They help to define a set of standards that programmers can follow when working collectively. When programmers choose a certain framework, they adopt the specific tools and methodologies associated with that framework. This also means they must be mindful of your choice, as they may end up with processes that don't fit the needs of their project or the developers involved. | Metadata modeling tool |
| 2. | Security Implementations | IAM Controls and Encryptions are implemented to improve security of the application. | SHA-256, Encryptions, IBM Cognos security |
| 3. | Scalable Architecture | Scalable operations are implemented using APIs like HTTP, HTTPS. | Planning Services, API Gateway |
| 4. | Availability | To ensure high availability and optimal service, the load balancer performs continual health checks of each server in the cluster, using probes to determine its eligibility for requests. Also it can user FIFO method to serve user based on their login | Server Load Balancers, FIFO |
| 5. | Performance | Performance of the system is increased using caching methodology. Caching mainly helps in storing the data without any data loss. | Caching |