**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

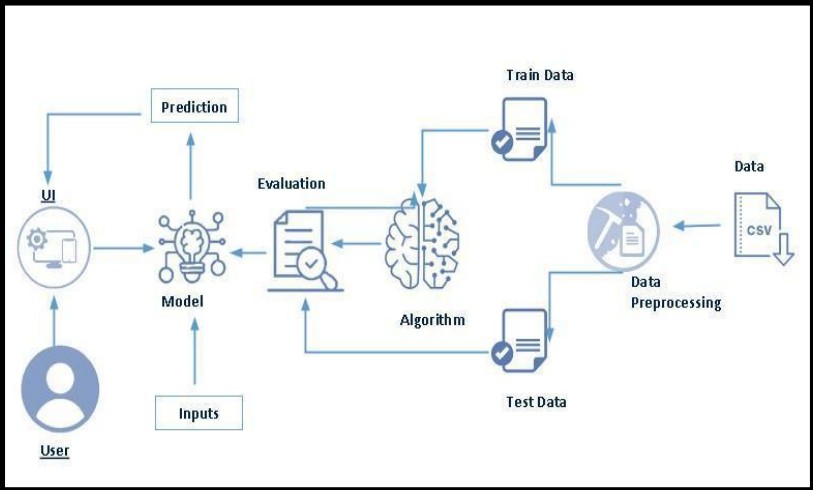
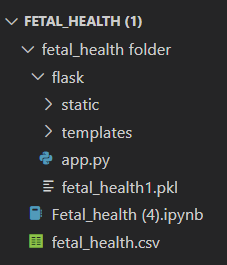
|  |  |
| --- | --- |
| Date | 03 October 2022 |
| Team ID | PNT2022TMIDxxxxxx |
| Project Name | Project - xxx |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

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

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**Technical Architecture:**

**+---------------------------------+-----------------+**



[This Photo](https://www.pngall.com/python-programming-language-png/) by Unknown Author is licensed under [CC BY-NC](https://creativecommons.org/licenses/by-nc/3.0/)

**| Data Collection & Preprocessing |**

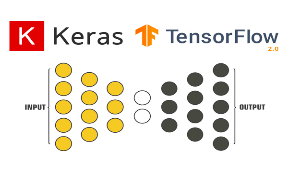
**| Python (Pandas, NumPy) |**

**+---------------------------------+-----------------+**

**|**

**v**

**+-------------------------------------------+**



[This Photo](https://dwctod.github.io/passages/Keras-vs-tf-keras-%E5%9C%A8TensorFlow-2-0%E4%B8%AD%E6%9C%89%E4%BB%80%E4%B9%88%E5%8C%BA%E5%88%AB/) by Unknown Author is licensed under [CC BY-SA-NC](https://creativecommons.org/licenses/by-nc-sa/3.0/)

**| ML Model Development |**

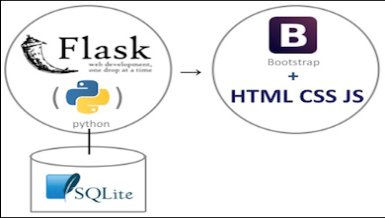
**| Python (TensorFlow, Keras) |**

**+--------------------------------------------+**

**|**

**v**

**+---------------------------------------------+**



[This Photo](https://devopedia.org/flask) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**| Web Application Interface |**

**| Flask (Python), HTML/CSS/JS |**

**+---------------------------------------------+**

**|**

**|**

**|**

**V**

**+---------------------------------------------------+**



[This Photo](https://smlpoints.com/notes-hiskio-cource-aws-gcp-aws-intro.html) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

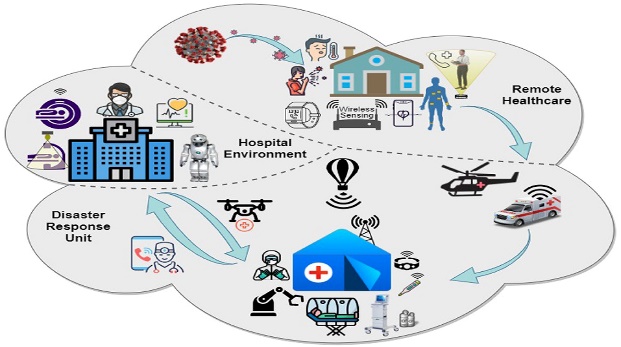
**| Cloud-based Storage |**

**| Amazon S3, Google Cloud Storage |**

**+---------------------------------------------------+**

**|**

**v**



[This Photo](https://www.frontiersin.org/articles/10.3389/frcmn.2020.610879/full) by Unknown Author is licensed under [CC BY](https://creativecommons.org/licenses/by/3.0/)

**+--------------------------------------------------+**

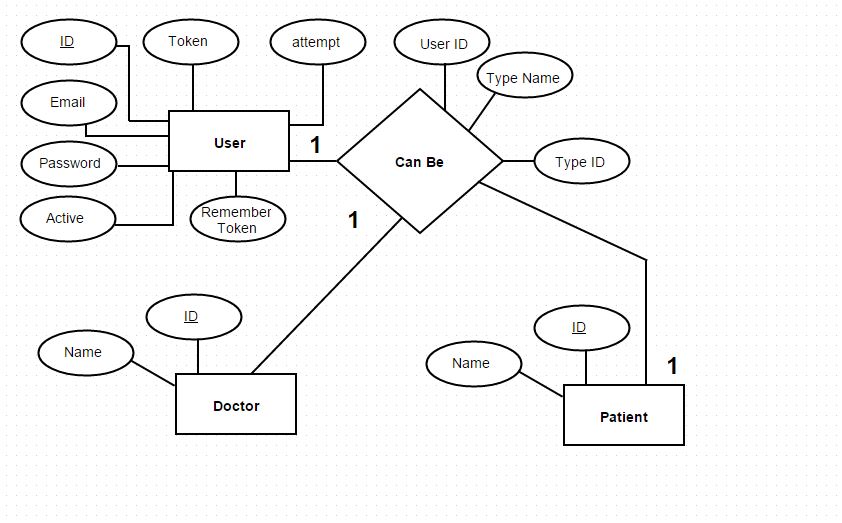


[This Photo](https://fr.wikipedia.org/wiki/Kaggle) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**| External Healthcare Data Sources |**

**| Hospital Databases, APIs |**

**+--------------------------------------------------+**



[This Photo](http://dba.stackexchange.com/questions/117786/help-me-to-create-correct-er-diagram) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**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, Flask |
| 2. | Application Logic-1 | Logic for a process in the application | Python |
| 3. | Application Logic-2 | Logic for a process in the application | **Python Libraries**:Pandas and NumPy,Scikit-learn  **Machine Learning Frameworks**:  TensorFlow or PyTorch |
| 4. | Application Logic-3 | Logic for a process in the application | **Python Libraries**:Pandas and NumPy,Scikit-learn, Flask or Django, AWS (Amazon Web Services) or Google Cloud Platform: |
| 5. | Database | Data Type, Configurations etc. | MySQL, NoSQL,MangoDB etc. |
| 6. | Cloud Database | Database Service on Cloud | IBM AWS cloud architecture, Google cloud services etc. |
| 7. | File Storage | File storage requirements | IBM Block Storage, Amazon cloud storage(S3) or Other Storage Service or Local Filesystem |
| 8. | External API-1 | Purpose of External API used in the application | Medical Database API: |
| 9. | External API-2 | Purpose of External API used in the application | Medical Imaging API: |
| 10. | Machine Learning Model | Purpose of Machine Learning Model | Classification models ,CNN, Random Forest etc |
| 11. | Infrastructure (Server / Cloud) | Application Deployment on Local System / Cloud Local **Server Configuration**: Utilizing services from cloud providers like Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure.  **Deployment Environment**: Deploying the application and machine learning models on cloud servers or platforms (e.g., AWS EC2, Google Compute Engine, Azure App Service). | Local, Cloud Foundry, Kubernetes.  AWS EC2, Google Compute Engine, Azure App Service,etc |

**Table-2: Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | List the open-source frameworks used | Technology of Opensource framework  TensorFlow, Keras, OpenAI, Scikit learn |
| 2. | Security Implementations | List all the security / access controls implemented, use of firewalls etc. | e.g. SHA-256, AES-256 Role-Based Access Control (RBAC), HTTPS/TLS),Encryptions, IAM Controls, OWASP, etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3 – tier, Micro-services) | Spring Boot (Java), Flask (Python), or Node.js for developing microservices. cloud services such as AWS ECS, Google Kubernetes Engine (GKE), or Azure Kubernetes Service (AKS) for scalable infrastructure provisioning and management. |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributed servers etc.) | Utilize technologies like NGINX, HAProxy, or cloud-based load balancers (AWS Elastic Load Balancing, Google Cloud Load Balancing).  AWS EC2, Google Cloud VM) AWS CloudFront. |
| 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:** Redis, Memcached  **Load Balancing:** NGINX, HAProxy, AWS Elastic Load Balancing;  **Browser Cache Control**: HTTP headers; **CDNs**: Cloudflare, AWS CloudFront, Akamai |

**References:**

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/) [**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture) [**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the**](https://medium.com/the)[**-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)