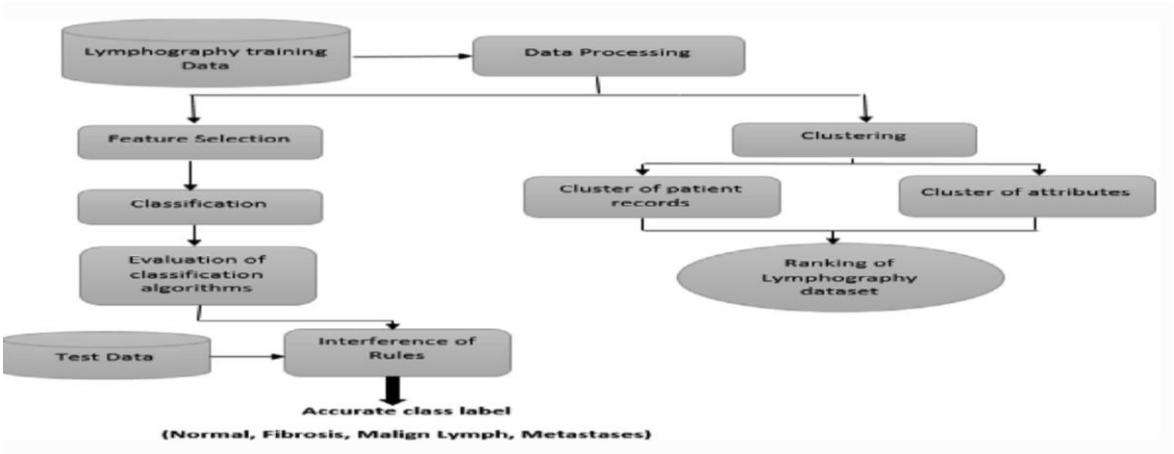
Technology Stack (Architecture & Stack)

Date	27th October 2023
Team ID	593213
Project Name	Lymphography Classification using ML
Maximum Marks	4 Marks

Technical Architecture:



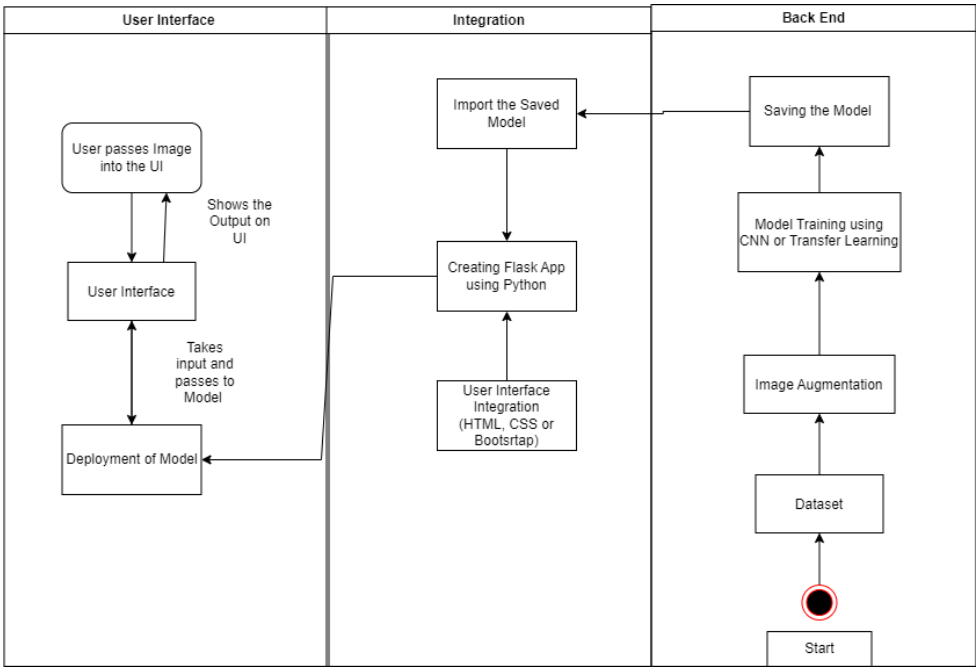


Table-1 : Components & Technologies:

Component	Description	Technology
1.Data Collection	Collecting lymphography images for training and testing.	• Medical imaging equipment (e.g., MRI, CT) • DICOM (Digital Imaging and Communications in Medicine) format for storage and retrieval of medical images • Data augmentation tools (e.g., OpenCV)
2.Data Preprocessing	Preparing data for model training and testing, including normalization, resizing, and data augmentation.	• Python (NumPy, Pandas, scikit-learn) • Image processing libraries (e.g., OpenCV)
3.Data Storage	Storing raw and preprocessed lymphography images for easy access and retrieval.	• Relational databases (e.g., PostgreSQL) • Distributed storage (e.g., Hadoop HDFS)
4.Machine Learning Model	Building and training a model to classify lymphography images.	• Convolutional Neural Networks (CNNs) • Deep learning frameworks (e.g., TensorFlow, PyTorch)
5.Model Evaluation	Evaluating model performance using validation data and metrics such as accuracy, precision, recall, and F1-score.	• Python (scikit-learn, TensorFlow, or PyTorch)
6.Model Deployment	Deploying the trained model to make predictions on new data.	• Web frameworks (e.g., Flask, Django) • Cloud services (e.g., AWS, Azure) for scalability
7.User Interface	Providing an interface for users to interact with the model, upload images, and view classification results.	• HTML, CSS, JavaScript for web applications • Front-end frameworks (e.g., React, Angular)
8.Security and Privacy	Implementing security measures to ensure data privacy and protect against unauthorized access.	• Authentication protocols (e.g., OAuth, JWT) • SSL/TLS for data encryption • Compliance with data protection regulations

9.Monitoring and Logging	Monitoring system health and logging activities, errors, and performance.	• Logging frameworks(e.g.Log4j, Logstash) • Monitoring tools
10.Scalability	Ensuring the system can handle increased load by distributing requests.	• Load balancers • Container orchestration (e.g., Kubernetes)

Table-2: Application Characteristics:

Characteristics	Description	Technology
1.Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2.Security Implementations	List all the security / access controls implemented,use of firewalls etc.	e.g.SHA-256, Encryptions, IAM Controls, OWASP etc.
3.Scalable Architecture	Justify the scalability of architecture (3 – tier,Micro-services)	Technology used
4.Availability	Justify the availability of application (e.g.use of load balancers, distributed servers etc.)	Technology used
5.Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used