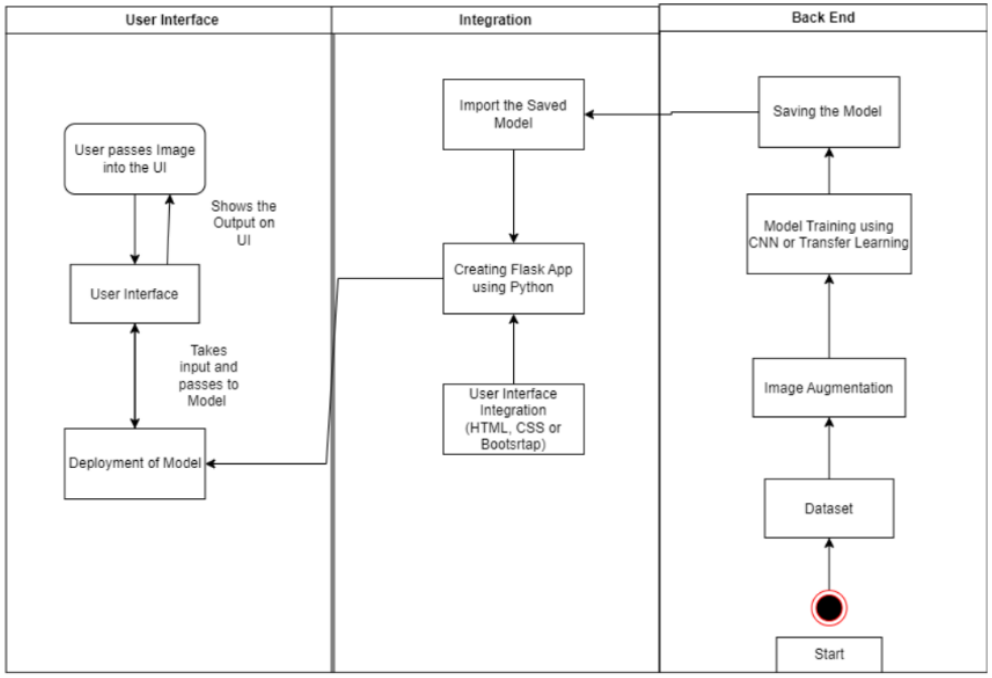
Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	31 October 2023
Team ID	Team-593067
Project Name	Horology 2.0: Forecasting The Future of Smartwatch Prices
Maximum Marks	4 Marks

Technical Architecture:

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



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	Facilitates user interaction, which could be a Web UI, mobile app, or other interfaces.	HTML, CSS, JavaScript, React, Angular, or Vue.js for web-based UIs; Swift or Kotlin for mobile app development.
2.	Data Collection Module	Gathers data from various sources such as sales records, economic indicators, reviews, etc.	Python for scripting, APIs for data retrieval, web scraping tools like Beautiful Soup or Scrapy.
3.	Data Processing and Cleaning	Involves cleaning and preprocessing collected data, handling missing values, and standardizing formats.	Pandas, NumPy for data manipulation, and transformation; Apache Spark for big data processing.
4.	Machine Learning Model Development	Building models to forecast smartwatch prices.	Scikit-learn, TensorFlow, or PyTorch for machine learning; regression, time series models, or neural networks for forecasting.
5.	Model Evaluation and Validation	Ensures model accuracy and reliability before deployment.	Cross-validation techniques, performance metrics (RMSE, MAE), and statistical methods for model validation.
6.	Performance and Optimization	Ensures the system's scalability and performance under varying loads. Hyperparameter tuning	Scaling techniques like Kubernetes for managing containerized applications, load balancing for optimized performance.
7.	Prediction	Predicts the value based on the user input.	ML algorithm, python libraries.

8.	Visualization and Reporting	Presents forecasted data and insights to users.	Visualization tools like Matplotlib, Seaborn, Tableau, or Power BI for creating interactive charts and dashboards
9.	Real-time Data update	Processes incoming real-time data for immediate analysis and updates.	Streaming technologies like Apache Kafka or AWS Kinesis for handling continuous streams of data.
10	Model Deployment	Deploys trained models to a production environment.	Containerization using Docker, deployment frameworks like Flask, Fast API, or serverless deployment using AWS Lambda. deployment using Docker and Flask or Fast API.
11.	Feedback Mechanism	Collects user feedback for model improvement and system enhancements.	Implementation of user feedback loops and data collection mechanisms.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Data-Driven Approach	Relies on data analysis and statistical modeling to forecast smartwatch prices.	Python for data processing, Pandas, NumPy for numerical computations, and SQL for data querying.
2.	Adaptability and integration	Adapts to changing market conditions, consumer behavior, and new data inputs.	Continuous learning models, adaptive machine learning algorithms, and data streaming for model retraining.
3.	Accuracy and Precision	Focuses on producing accurate and precise forecasts to support decision-making.	Fine-tuning models, hyperparameter optimization, and robust evaluation techniques for accuracy assessment.

4.	Automation	Minimizes manual intervention, automates data processing, and model retraining.	Automated pipelines using tools like Apache Airflow, CI/CD for model deployment, and monitoring systems.
5.	Security and Compliance	Ensures data security and compliance with privacy regulations during data handling and processing.	Encryption methods, secure data transmission (SSL/TLS), and compliance frameworks for data handling.