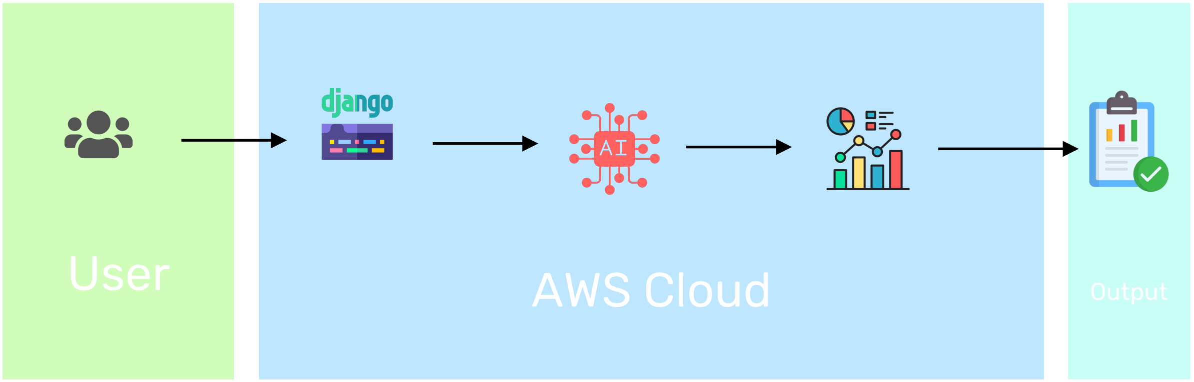
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	08 November 2023
Team Id	Team-593195
Project Name	Dog Breed Identification using Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

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

Dog Breed Identification Using Transfer Learning



1. **AI Model Build using Transfer Learning Process:**
   - **Data Collection**: Gather relevant data for the target domain.
   - **Preprocessing**: Clean, preprocess, and augment the data to improve model performance.
   - **Model Selection**: Choose a pre-trained model suitable for transfer learning.
   - **Feature Extraction**: Extract relevant features from the pre-trained model.
   - **Fine-tuning**: Fine-tune the pre-trained model using the domain-specific dataset.
   - **Evaluation**: Assess the model's performance and make necessary adjustments.
   - **Deployment**: Prepare the model for deployment within the Django website.

2. **Infrastructural Demarcation:**
   - **Local Infrastructure**: Utilize local servers or machines for model training and inference.
   - **Cloud Infrastructure**: Consider using cloud platforms like AWS, GCP, or Azure for scalability and efficient resource management.

3. **External Interfaces:**
   - **Django REST API**: Expose the AI model's functionalities through a Django REST API for seamless integration with the website.
   - **Third-Party APIs**: Integrate third-party APIs for additional functionalities, such as payment gateways or authentication services.

4. **Data Storage Components/Services:**
   - **Database Management System**: Utilize Django's built-in ORM for efficient data storage and retrieval.
   - **Cloud Storage**: Employ cloud-based storage services for managing large datasets and model artefacts.

5. **Interface to Machine Learning Models:**

- **Django Model Integration**: Incorporate the AI model within the Django application's backend logic.
- **REST API Endpoints**: Create dedicated endpoints within the Django app to handle requests and responses related to the AI model.
- **Model Inference**: Implement model inference logic to process incoming data and generate predictions or classifications.
- **Data Visualization**: Integrate tools for visualizing model outputs or predictions within the website's front end for user interaction.

Table-1 : Components & Technologies:

Sl No.	Component	Description	Technology
1	User interface	We’re looking to build UIs using Django. This will be basically the web app deployed on AWS servers fro users to interact with	HTML, CSS, Django Templates
2	Machine Learning Model	We’ll build machine learning model by using the technique called transfer learning	Python, Tensorflow, VGG19
3	AWS Cloud host	We'll use AWS Cloud to deploy our app and host the website	AWS S3
4	AWS Compute Engine	We’ll use AWS EC2 instance for the deployment of ML Model	AWS EC2

Table-2: Application Characteristics:

Sl No.	Component	Description	Technology
1	Open-Source Frameworks	We'll use Django as open source fromework to build our webapp	Django
2	Security Implementations	We’ll enable security feature for the to and fro data sharing of the user like image used for prediction as it may be of his/her own dog picture	SHA-256, Encryptions,
3	Scalable Architecture	Architecture that is used will be scalable, will be build as modules and all the modules will be loosely coupled for better maintenance and scalable	Clean Architecture

4	Availability	To ensure availability w’ll use AWS Cloud service, that is also know for the robustness and reliability	AWS
5	Performance	We'll leverage all the better features from the AWS Cloud, like caching region based S3 buckets and more.	AWS CDN Services