**Project**

**PicAi**

**Aim: PicAi is a web application that utilizes generative adversarial networks to manipulate or edit images. It incorporates state-of-the-art technologies such as ModNet for removing the background from portrait images, enhancers like GPGAN and RealESRGAN to improve image quality, and filters like Anime and Arcane to transfer image styles.**

**COMPONENTS AND FEATURES USED IN THIS PROJECT**

* **TOOLS USED:**

**1. FastAPI (For backend server in Python)**

**2. HTML, CSS, and Vanilla JavaScript (For frontend)**

**3. Nginx server (As a reverse proxy server)**

**4. Docker (Containerization tool to package all software)**

**5. Certbot (SSL encryption for the website)**

**6. PyTorch (To fine-tune the GAN models)**

**7. ONNX (To quantize the model, enabling it to run on edge devices)**

**FEATURES**

**MACHINE LEARNING MODELS**

**1.MODNet**

ModNet is a real-time portrait matting model that requires only RGB image input. It enables the removal of backgrounds from any portrait image.

**2.Enhancers (GFPGAN, RealESRGAN, UpCunet)**

These networks utilize generative adversarial networks to enhance image quality by simultaneously upscaling the resolution.

**3. Anime and Arcane Filters**

* **Anime Filter**: AnimeGanV2 is a network that uses generative adversarial networks to transfer the style of a real image to an anime style.
* **Arcane Filter:** This model also uses generative adversarial networks to transfer the style of a real image to an arcane style. To enhance the overall image quality, GFPGAN is employed to improve facial structure while preserving facial expressions and features.

**CODEGEN: LLM DRIVEN CODING**

**[Collaboration project]**

This project explores the impact of replitv3 3B model, a language model, on code generation. By training it extensively on diverse code and programming languages, we aim to enhance its understanding of coding paradigms and syntax. Through fine-tuning, the model gains a deep grasp of programming context, allowing it to predict coding patterns.

**COMPONENTS AND FEATURES CREATED IN THIS PROJECT**

* **TOOLS USED:**

**1. FastAPI [Backend Server]**

**2. replitv3 3B [Code Instruct model with ggml quantization]**

**3. Docker [For conternization with watchtower enabled]**

**4. github action [for ci/cd and fast release of code]**

**5. React JS [dashboard]**

**6. HTML CSS & VANILLA JS [for user interface to use code instruct model]**

**7. Redis [Storing Chat history] (collaboration with Heta Vyas)**

**8. OAuth & JWT For Secure login & navigation**

**9. Google Sign in**

**Features**

**Machine Learning**

**1. Replitv3 3B**

Replit Code v3is a 3B parameter Causal Language Model focused on Code Instruction.

For faster inference we employed quanitzation method like ggml to provide both good accuracy and faster inference.

**2. Google Sign In, JWT & OAuth**

JWT or Json Web Token it is employe to provide secure access to the web application.

OAuth2.0 to provide secure authentication for the user

**3. Redis**

Redis is employed to store all the chat history to the database.

**# SKILLS**

**1. Python [intermediate]**

**2. FastAPI ASGI Server [Intermediate]**

**4. Machine learning [Supervised & Semi-Supervised]**

**Framework known**

**1 Pytorch [Intermediate]**

**2.tensorflow [Beginner]**

**3. onnx [Beginner]**

**6. Computer Vision**

**Framework known**

**1. OpenCV**

**7. Docker & docker-compose.yml [intermediate]**

**8. Git & Github action (CI/CD) [intermediate]**

**9. Cloud Deployment**

**Services known**

**1. AWS [Amazon Web Service]**

**2. Azure**

**3. Google Cloud Platform**

**10. Devops**

**Devops tools**

**1. Jenkins**

**2. Docker**

**3. Jira**

**COLLABSTR CLONE**

**AIM: FIND AND HIRE INFLUENCER**

**THE PLATFORM BASED ON INFLUENCER MARKETING WHERE BRAND FIND THEIR DESIRED NICHES AND BID ON THE INFLUENCER.**

**COMPONENTS AND FEATURES USED IN THIS PROJECT**

**1. FastAPI [For Server]**

**2. ReactJS [For Frontend]**

**3. Docker**

**4. JWT, OAuth, Google Sign in, Location API.**

**5. SMTP SERVER [FOR OTP]**