CreativeCollab – Image Synthesis

Core Technologies

1. Deep Learning Framework

PyTorch → this is the backbone for training and running SPADE (Semantic Image Synthesis).

I am using PyTorch for:

- Handling datasets (images + segmentation maps).
- Defining and training the SPADE generator/discriminator.
- Running inference (turning segmentation maps into photo-realistic landscapes).

2. Dataset Handling

- COCO-Stuff dataset → you're already preparing it (images + pixel-level masks).
- Custom preprocessing → remapping COCO's 182 classes → your chosen ~20 landscape classes (sky, mountain, river, etc.).

Tools: Python, PIL, NumPy for processing masks and images.

3. Model Architecture

- SPADE (Spatially-Adaptive Normalization) → GAN-based architecture.
- Generator: takes segmentation map + noise → synthesizes realistic image.
- Papers: "Semantic Image Synthesis with Spatially-Adaptive Normalization (SPADE)".

4. Training & Experimentation

- Google Colab / Local GPU (NVIDIA) → for training your model.
- Mixed-precision training with PyTorch AMP → faster + memory efficient.

5. Frontend (User Interaction)

- React.js → for the sketching field where users can draw segmentation maps.
- User draws → segmentation map is generated (color-coded regions).
- This gets sent to the backend.

6. Backend (Inference API)

- Flask / FastAPI → wraps your trained model as a REST API.
- User uploads segmentation → model generates realistic landscape → sends back to frontend.

7. Deployment

- Backend: FastAPI + Docker (hosted on cloud, e.g., AWS, GCP).
- Frontend: React.

Tools Summary

Python + PyTorch → model training & inference.

PIL / NumPy / OpenCV → preprocessing masks & images.

React + Canvas libraries → sketching tool.

Flask/FastAPI → backend API for serving model.

Colab/Kaggle/Local GPU → training environment.

Docker → packaging model for deployment.