ClimateSync Modeler

1. Predict the ecosystem health score.
2. Predict potential productivity changes.
3. Perform detailed risk assessments and suggest mitigation strategies.

Setup Instructions

1. **Clone the repository:**

```
git clone <repository_url>
cd <repository_folder>
```

2. **Install dependencies:** Ensure you have Python installed, then run:

```
pip install -r requirements.txt
```

3. **Ensure required files are present:**
    a. climate_model.pkl (trained model file)
    b. features_list.txt (list of required features)
4. **Run the Flask app:**

```
python app.py
```

The application will run on http://127.0.0.1:5000 by default.

Endpoints

1. Predict Ecosystem Health Score

Endpoint: /predict

Method: POST

Description: Predicts the ecosystem health score based on environmental and operational factors.

Request Body: A JSON object containing the necessary features. Example:

```
{
   "Temperature_Anomaly": 1.2,
   "Sea_Surface_Temperature": 25.5,
   "Ocean_Acidity": 8.1,
   "Sea_Level_Rise": 0.3,
   "Salinity": 35,
   "Dissolved_Oxygen": 6.8,
   "Species_Diversity": 120,
   "Aquaculture_Production": 5000,
   "Climate_Zone": "Tropical",
   "Region": "Indian"
}
```

2. Predict Productivity

Endpoint: /predict-productivity

Method: POST

Description: Predicts potential productivity changes based on environmental factors.

Request Body: A JSON object containing the relevant factors. Example:

```
{
  "input_factors": {
    "Temperature": 28,
    "Salinity": 12,
    "pH": 7.2,
    "Dissolved Oxygen": 8
  }
}
```

3. Risk Assessment

Endpoint: /risk-assessment

Method: POST

Description: Performs a risk assessment and suggests mitigation strategies based on given risk factors.

Request Body: A JSON object containing the risk factors. Example:

```
{
  "risk_factors": {
    "Temperature": 33,
    "Salinity": 3,
    "pH": 9,
    "Dissolved Oxygen": 2
  }
}
```