Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **TravelSafe - AI Trip Risk Checker**. The project is a web application that allows users to input travel destinations and receive machine-learning-based risk predictions using recent data. The design leverages Python, Flask, and a simple ML model to deliver practical travel safety insights.

---

1. System Components

Component	Description	Key Responsibilities
Web UI	Frontend for user input and result display	Collect destination, show results
Flask API Server	Backend REST API	Handle requests, invoke ML model
ML Risk Predictor	Machine learning model/service	Predict risk for destination
Data Source	Recent travel/safety data (static or API)	Provide input features for ML model

---

2. Class/Interface Overview

Class/Module	Description	Key Methods/Attributes
`app.py`	Flask app entrypoint	`@app.route('/predict')`
`RiskPredictor`	ML model wrapper	`predict(destination: str) -> dict`
`DataFetcher`	Fetches recent data	`get_features(destination: str)`

Relationships:

- `app.py` uses `RiskPredictor` and `DataFetcher` to process requests.

Key Methods Example:

```python
class RiskPredictor:
    def predict(self, destination: str) -> dict:
        # Returns risk score and details
```

---

3. Data Structure Overview

Model/Schema	Fields/Attributes	Purpose
User Input	`destination: str`	User-provided location

Risk Prediction	`risk_score: float`, `risk_level: str`, `details: str`	Output to user
Feature Data	e.g., `crime_rate: float`, `weather_alert: bool`	ML model input features

---

4. Algorithms/Logic

Prediction Flow (Pseudocode):

```
def predict_risk(destination):
    features = DataFetcher.get_features(destination)
    risk_score = MLModel.predict(features)
    risk_level = categorize_risk(risk_score)
    return {"risk_score": risk_score, "risk_level": risk_level, "details": ...}
```

- **categorize_risk:** Maps score to “Low”, “Medium”, “High”.

---

5. Error Handling

Scenario	Handling Approach
Invalid destination input	Return 400 error with message
Data fetch failure	Return 503 error, log issue
ML model prediction error	Return 500 error, log stack trace
Unexpected server error	Return 500 error, generic message

End of Document