Customer Ticket Resolver Agent

## **Overview**

This project is a Flask-based customer support ticketing system that leverages machine learning to streamline and automate ticket handling. It incorporates two CSV files:

* **historical\_tickets.csv**: Contains resolved customer support tickets with their corresponding resolutions
* **new\_tickets.csv**: Contains incoming ticket queries requiring classification and resolution

The system satisfies the following functionalities:

* Maintains a dataset of historical tickets and resolutions (historical\_tickets.csv)
* Receives and processes new incoming tickets from new\_tickets.csv
* Classifies each new ticket into predefined categories using TF-IDF and Naive Bayes
* Retrieves semantically similar past tickets using SBERT
* Generates a draft resolution based on the most similar past solution
* Simulates an agent approval/edit workflow before finalizing the response
* Outputs the final reply and records the time taken for resolution

## **Machine Learning Components**

### **1. Category Predictor (category\_predictor.py):**

* **Purpose:** Predicts the category and subcategory of a ticket based on its description.
* **Model:** Utilizes a TF-IDF vectorizer combined with a Multinomial Naive Bayes classifier within a One-vs-Rest framework.
* **Training:** Trained on historical\_tickets.csv.

### **2. Resolution Suggester (resolution\_suggester.py):**

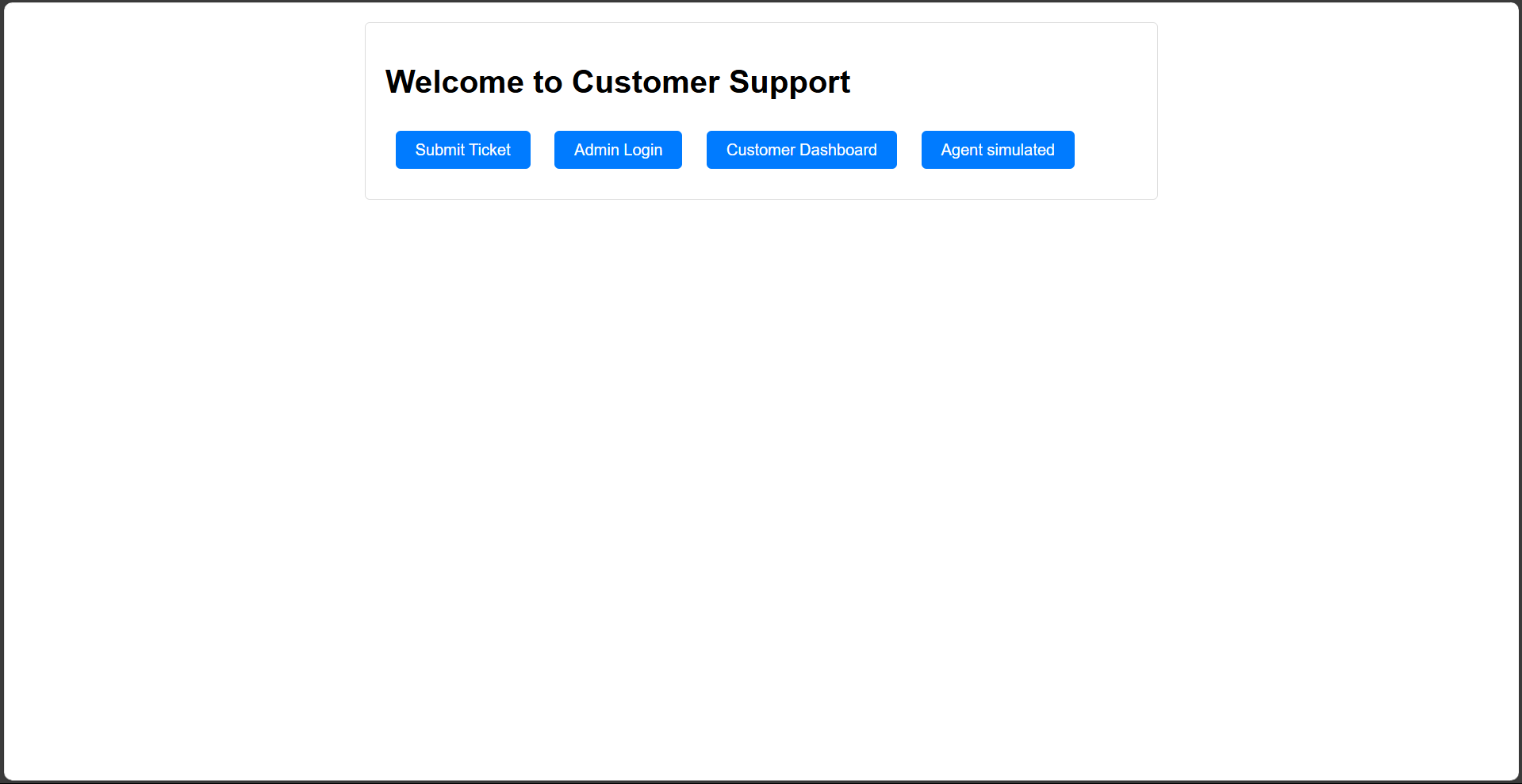
* **Purpose:** Suggests a resolution for a new ticket by finding the most similar past issue.
* **Model:** Employs Sentence-BERT (all-MiniLM-L6-v2) to compute embeddings and cosine similarity.

### **3. Auto-Approval System (auto\_approval.py):**

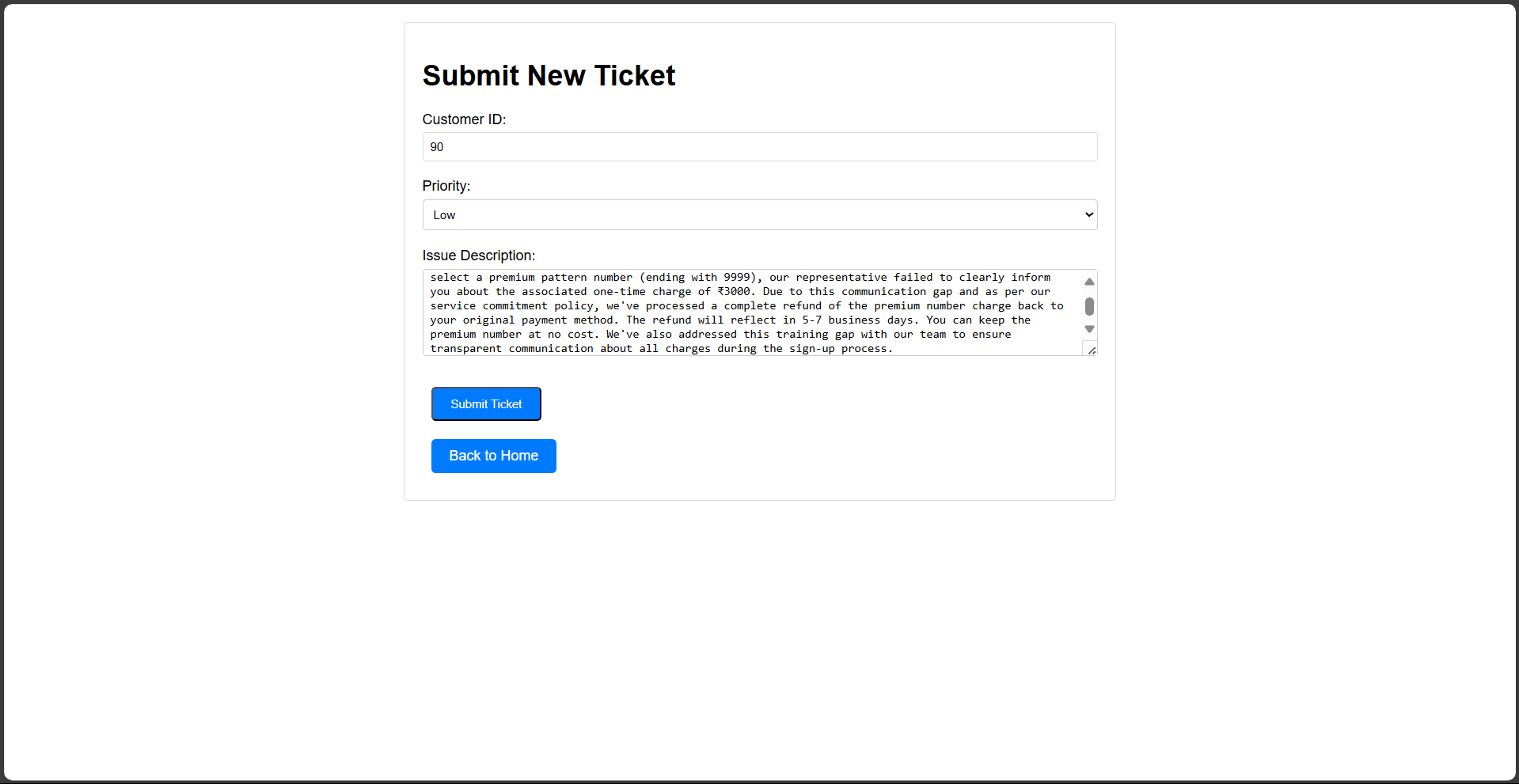
* **Purpose:** Determines whether a suggested resolution can be auto-approved based on similarity to past resolutions.
* **Mechanism:** Calculates cosine similarity between the new resolution and historical ones, comparing it against category-specific thresholds.

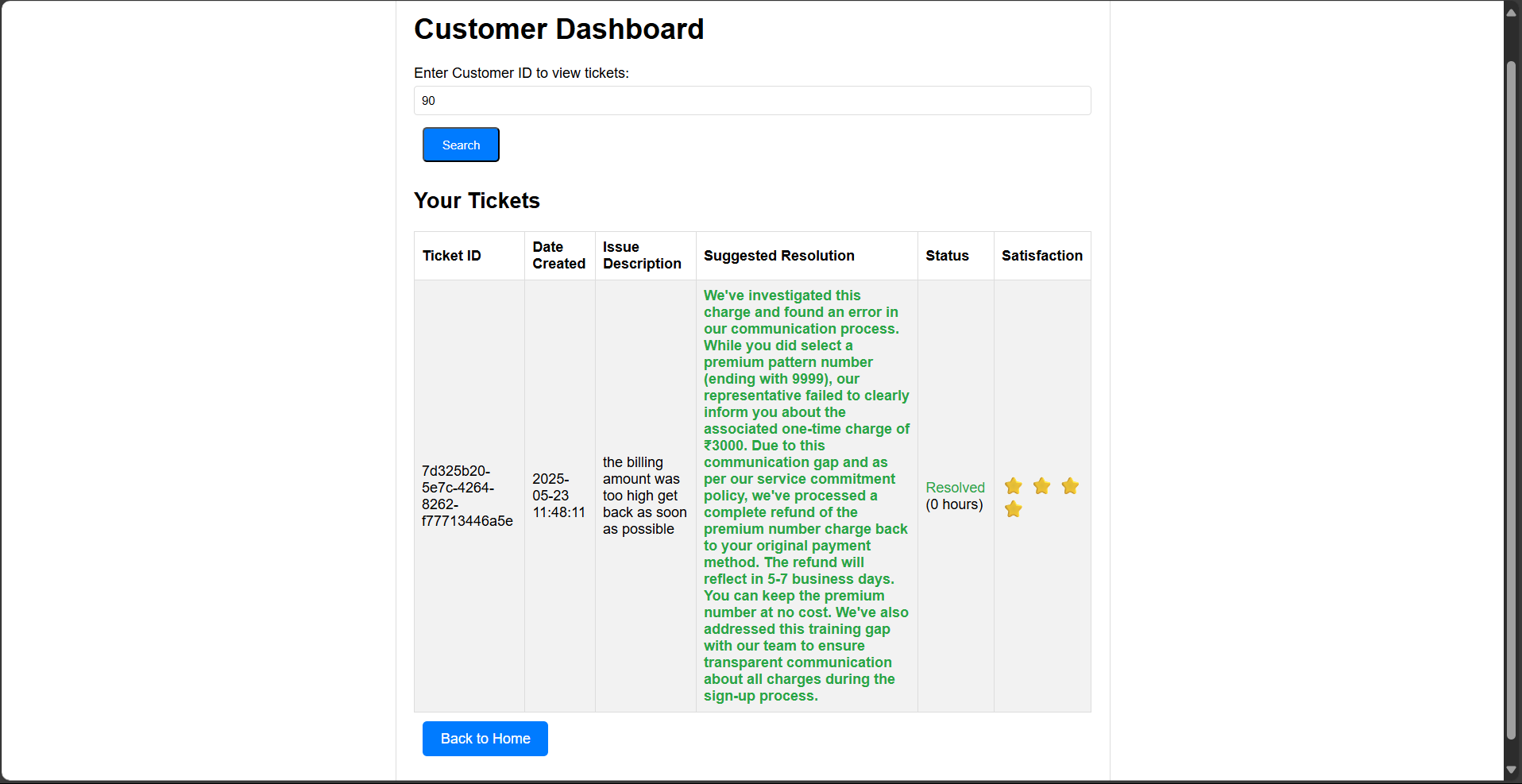
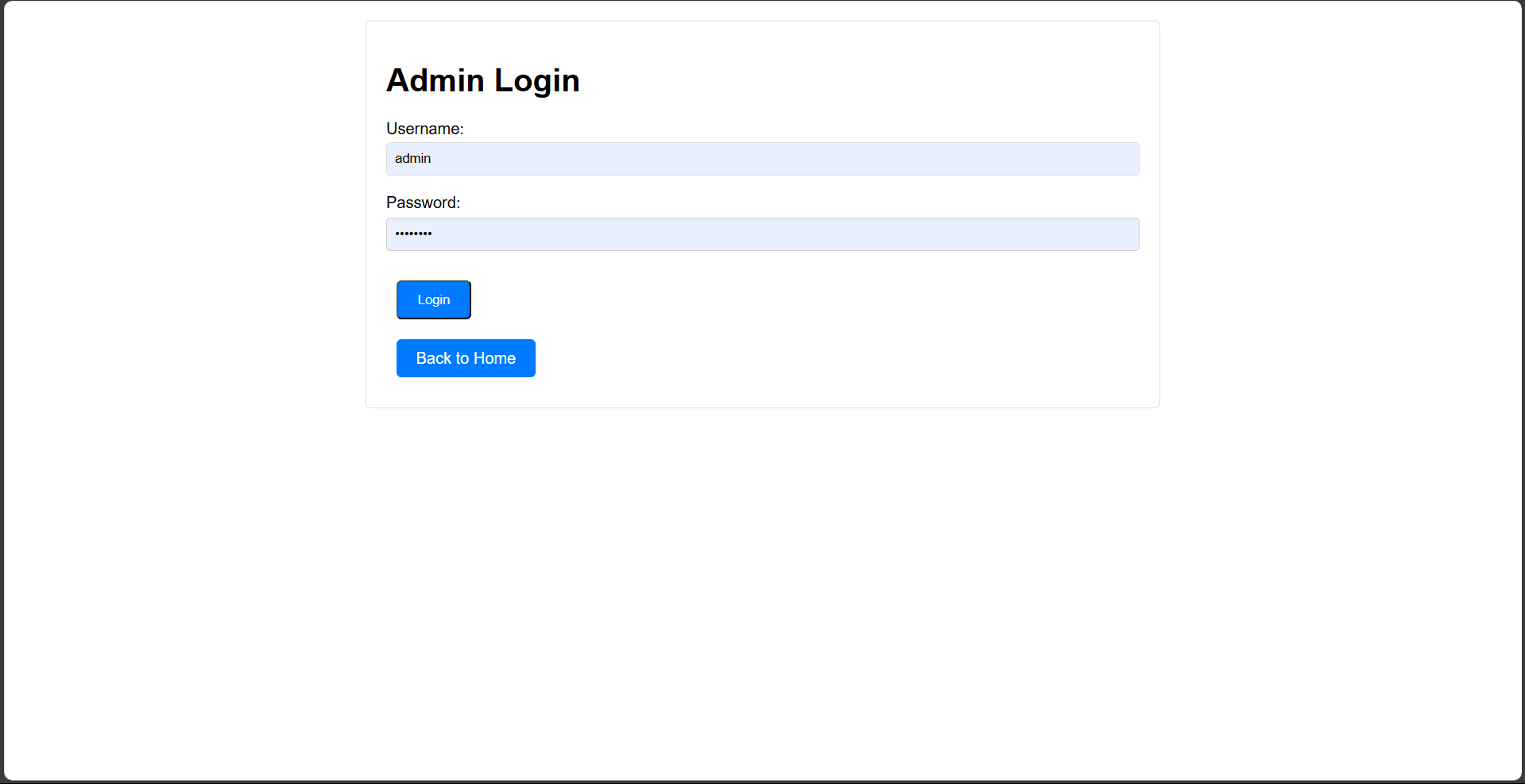
## **Application Routes**

* **/**: Home page.

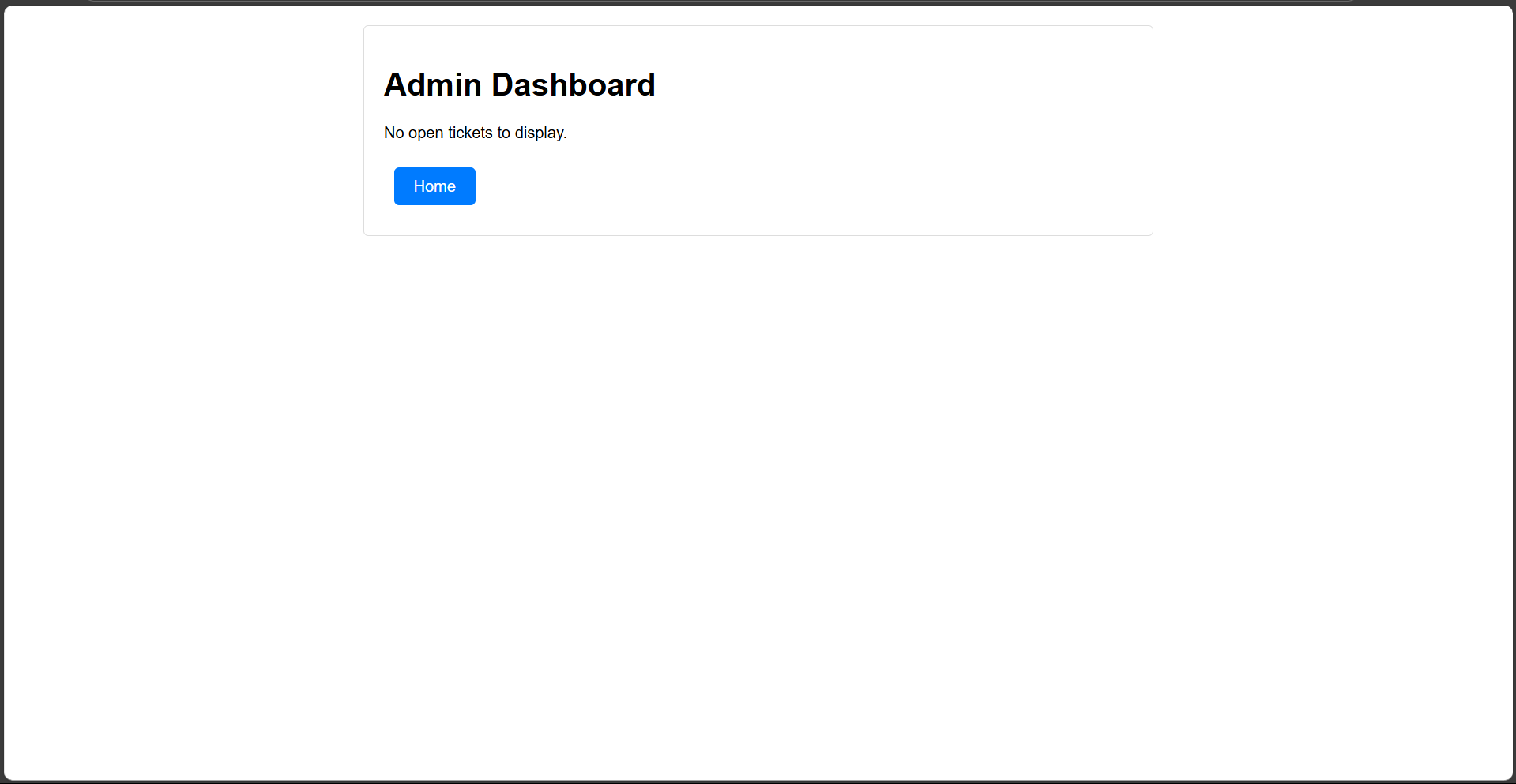


* **submit\_ticket**: Form for customers to submit new tickets.

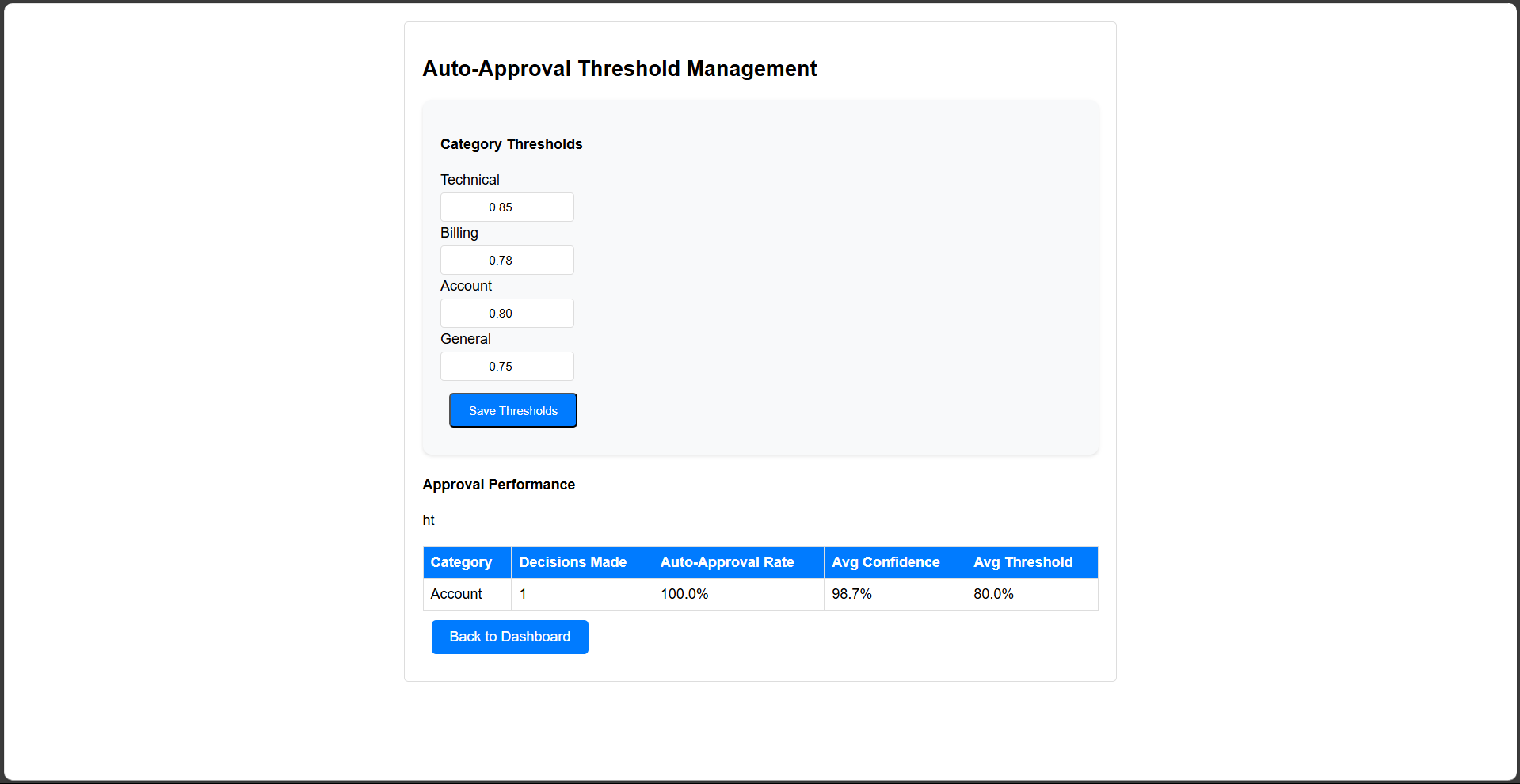


* **/customer/dashboard**: Dashboard displaying customer's tickets and their statuses.
* 
* **/admin/login**: Admin login page.  
  

**/admin/dashboard**: Admin dashboard showing unresolved tickets with suggested resolutions and auto-approval statuses.



* **/resolve/<ticket\_id>**: Endpoint for admins to manually resolve tickets.
* **/admin/thresholds**: Page for admins to view and adjust auto-approval thresholds per category.



## **Data Files**

* **data/historical\_tickets.csv**: Contains past tickets with fields like ticket\_id, customer\_id, date\_created, category, subcategory, priority, issue\_description, resolution, resolution\_time\_hours, customer\_satisfaction, and agent\_id.
* **data/tickets.csv**: Stores current tickets submitted through the application.

## **Admin Credentials**

* **Username:** admin
* **Password:** admin123

**Adjusting Auto-Approval Thresholds**

Admins can navigate to /admin/thresholds to view and modify the confidence thresholds required for auto-approving resolutions in each category. This allows fine-tuning the sensitivity of the auto-approval system based on category-specific requirements.