**Customer Support Ticketing System – Web Application**

# Project Overview

The Customer Support Ticketing System is a web-based platform that simplifies and streamlines customer support management. It enables customers to raise and track support tickets while allowing admins and support teams to view, assign, and resolve tickets efficiently. The system improves communication, ensures transparency, and provides insights into operational performance.

# Objectives

* Provide an intuitive platform for customers to raise and manage support tickets.
* Enable administrators and agents to assign, update, and resolve tickets efficiently.
* Offer detailed analytics and reporting for better decision-making.
* Maintain security, scalability, and usability following best practices.

# Features

User Portal:

* Secure Sign-Up & Login
* Profile management (update info, upload photo)
* Submit and track tickets- View ticket history with filters

Admin Portal:

* Dashboard with ticket filters (status, date, priority)
* Assign tickets to agents
* Update ticket status and add internal notes

Reports & Analytics:

* Ticket volume trends
* Resolution time analytics
* Agent performance metrics

# System Architecture

Frontend: HTML, CSS (Bootstrap/Tailwind), JavaScript

Backend: Python (Flask or Django)

Database: MySQL / PostgreSQL Version Control: Git & GitHub

Architecture Flow:

[User Browser] ■ [Flask Backend/API] ■ [MySQL Database]

■

■■■ [Admin Panel]

# Database Schema

The database includes tables for users, tickets, agents, and notes. Each ticket is linked to a user and optionally to an agent. Notes store internal communication related to a ticket.

The schema supports efficient querying, reporting, and role-based management.

# Installation Guide

# Clone the project **git clone https://github.com/Zuveriya-Tabassum/support-ticket.git**

**cd support-ticket**

# Create virtual environment

**venv\Scripts\activate** (Windows) or source

**venv/bin/activate** (Linux/Mac)

# Install dependencies

**pip install –r requirements.txt**

# Run the server

flask run or **python app.py** runserver

# Usage Instructions

User Side:

* Register or login
* Submit new support tickets- Track ticket status and view history

Admin Side:

* Login as admin/agent
* Manage and assign tickets
* Update status, add notes, and generate reports

# API Endpoints (Sample)

/signup – POST – Register user

/login – POST – Authenticate user

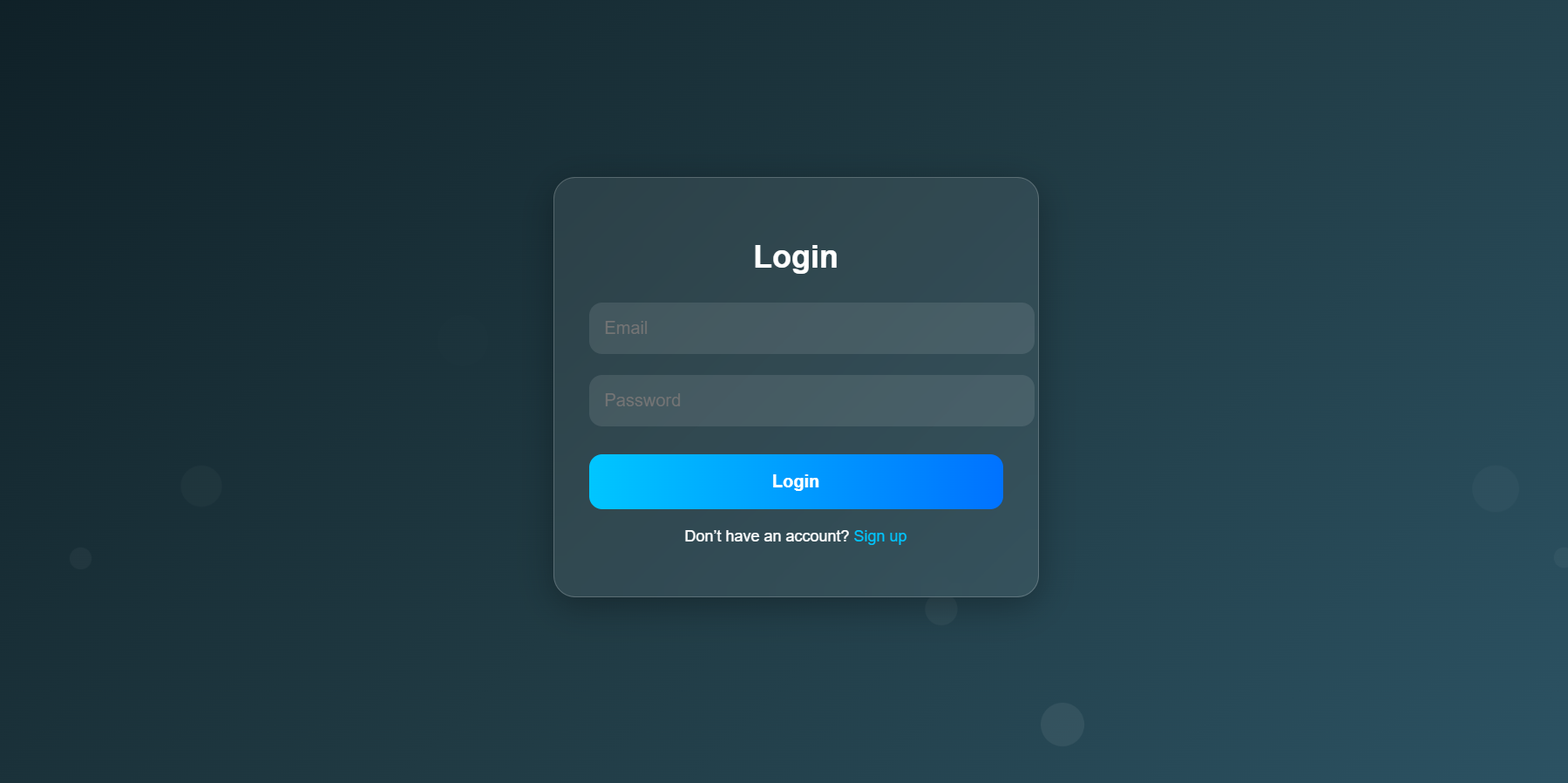
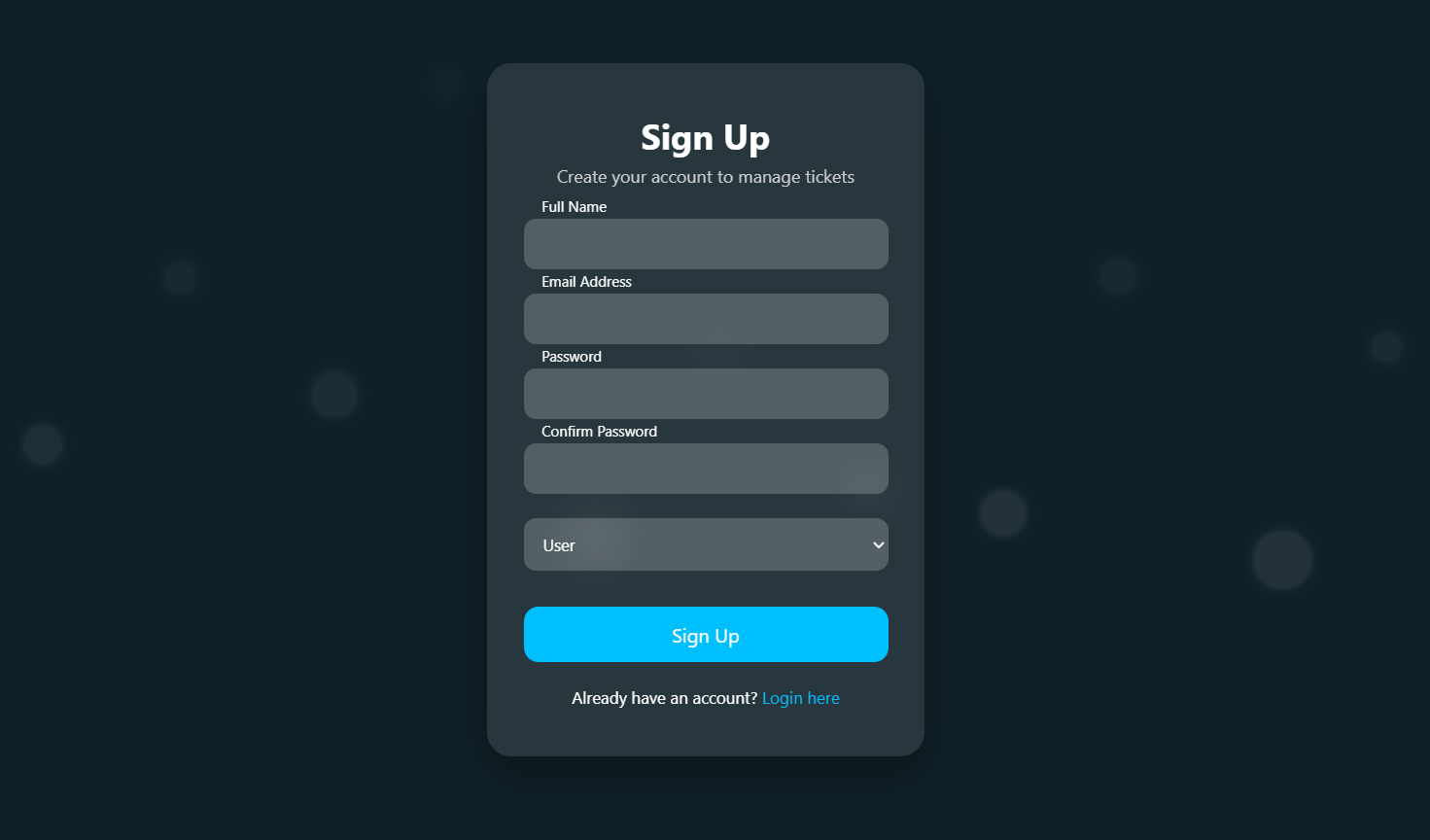
/tickets – GET/POST – List or submit tickets

/tickets/ – GET/PUT – View or update ticket

/admin/tickets/assign – POST – Assign ticket to agent

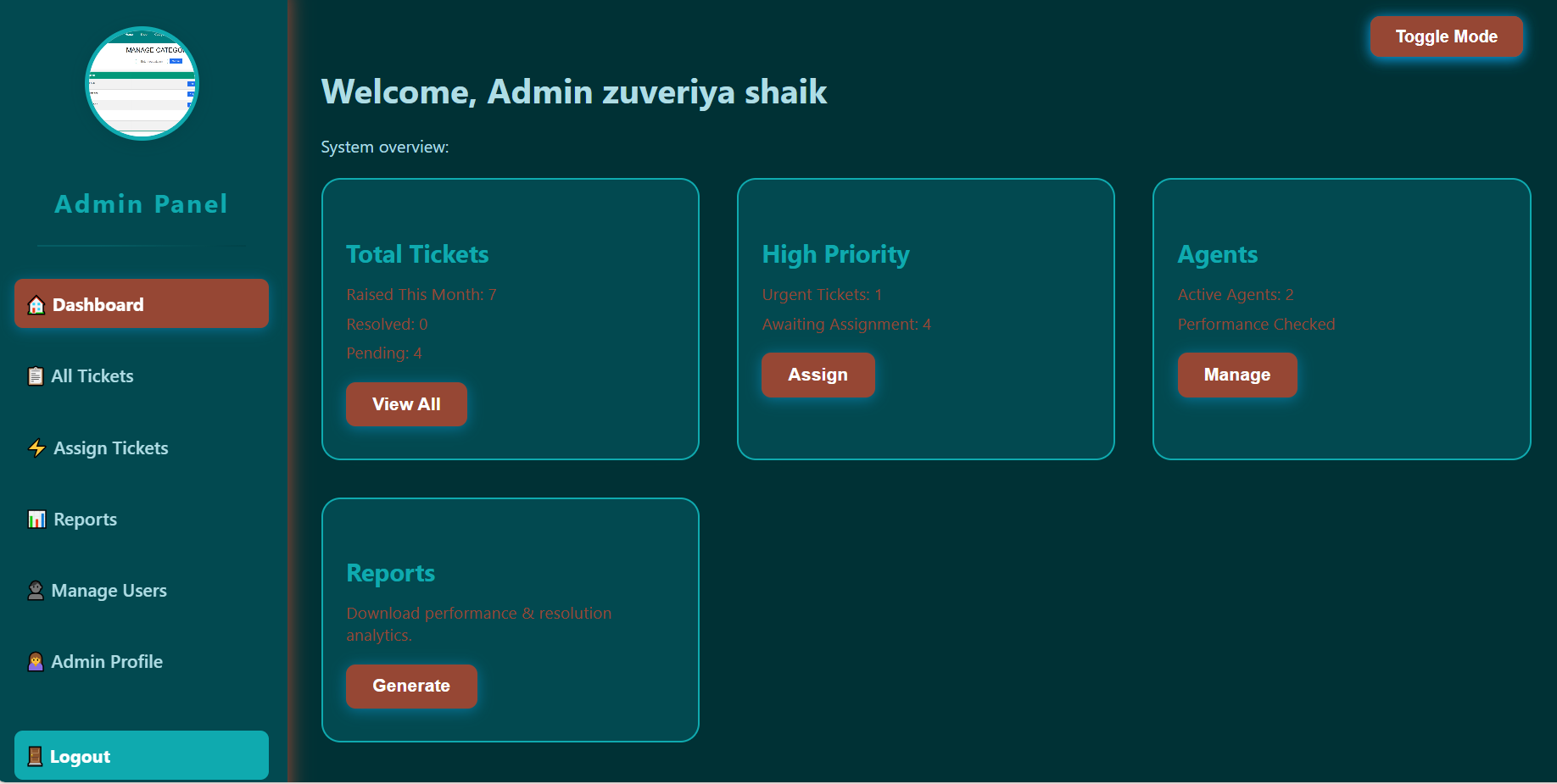
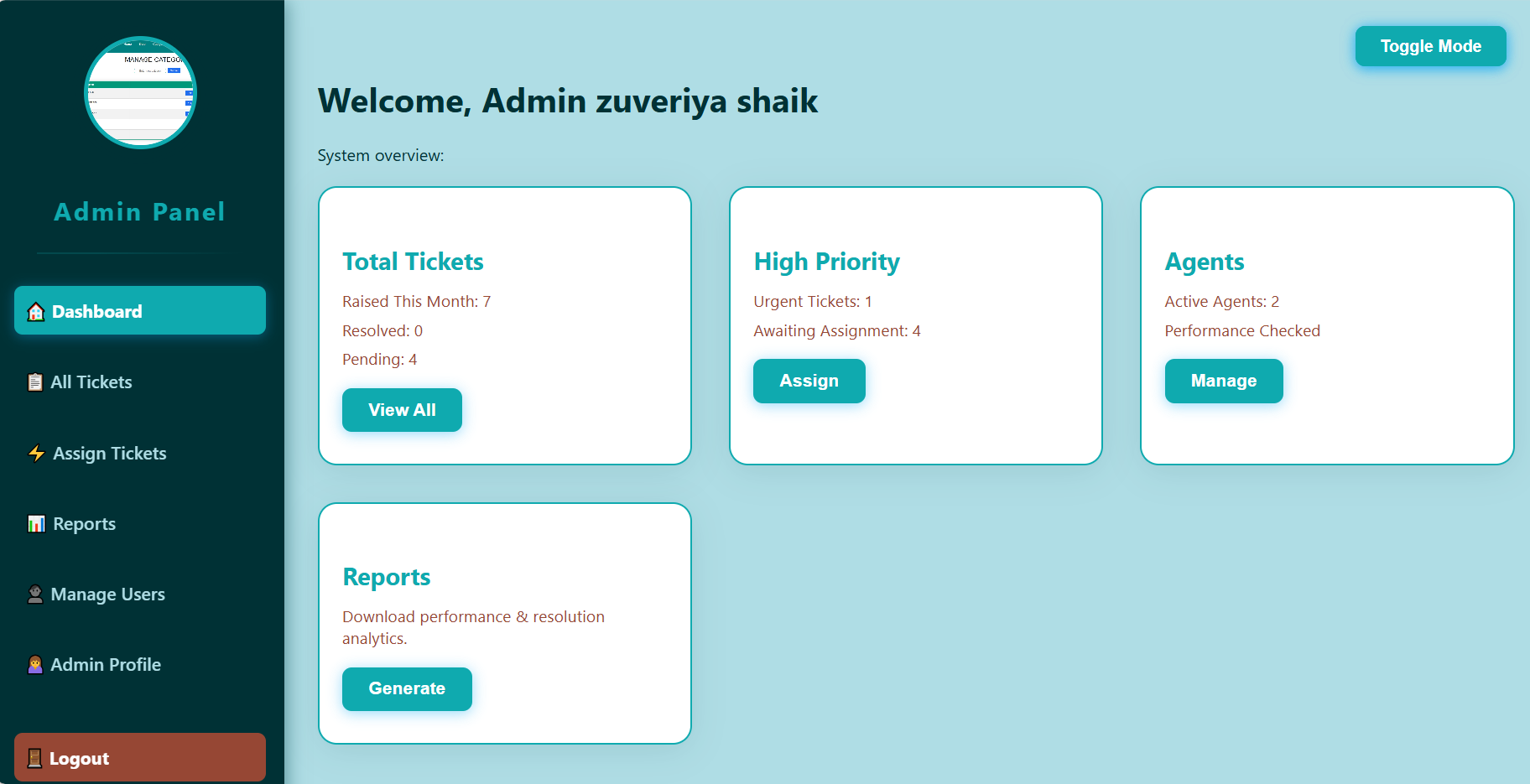
/admin/reports – GET – Fetch analytics data

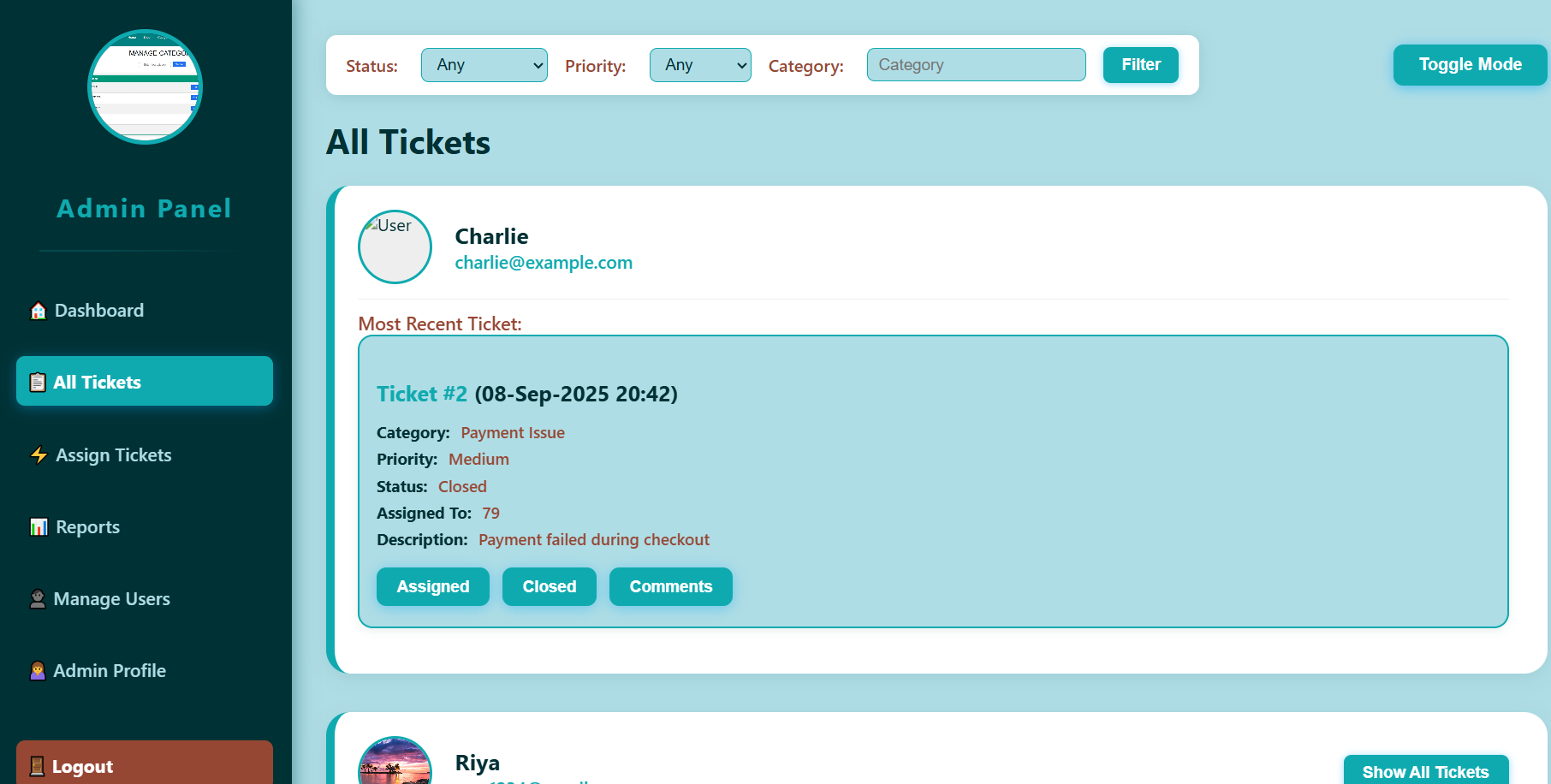
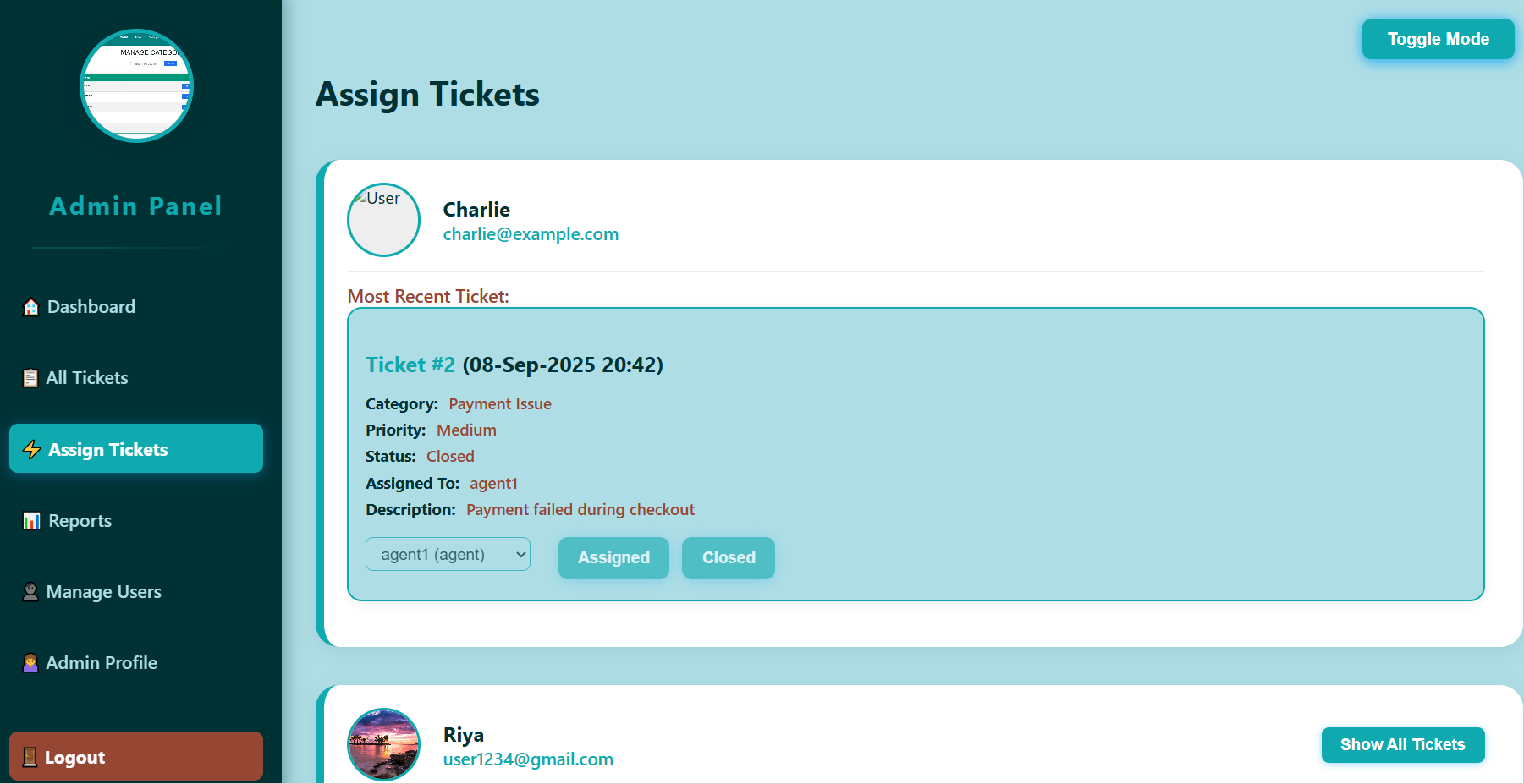
# Screenshots

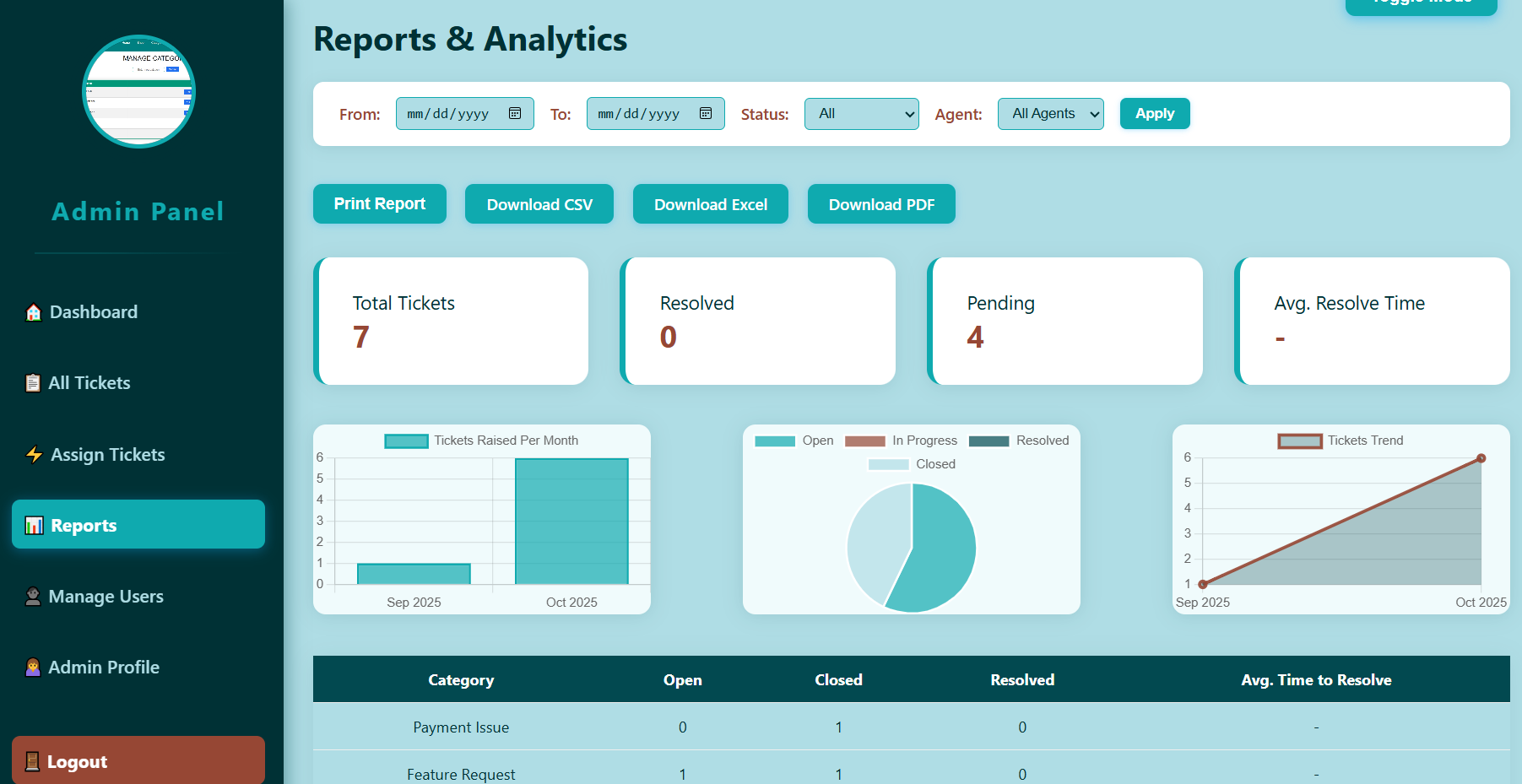
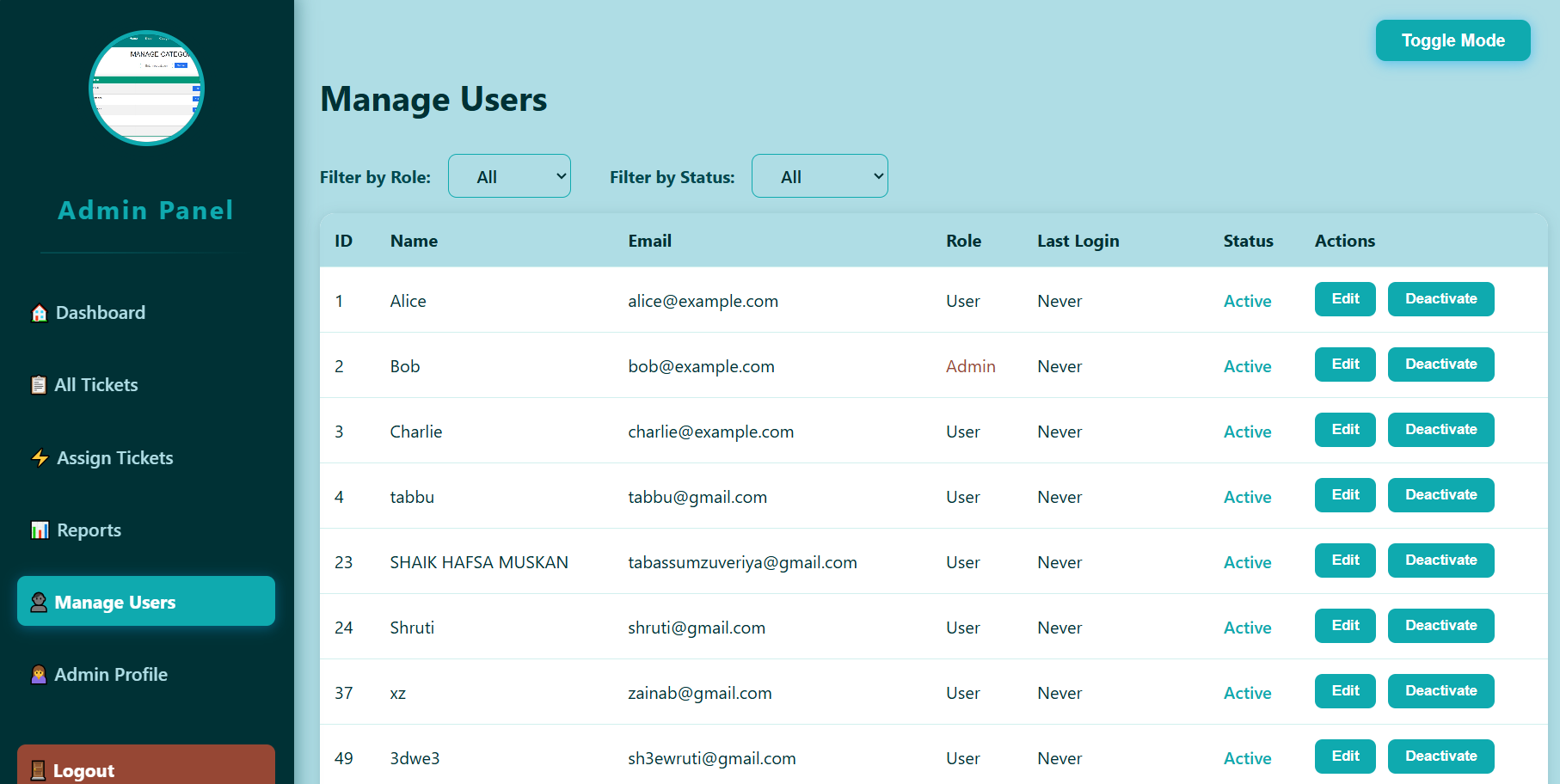
 

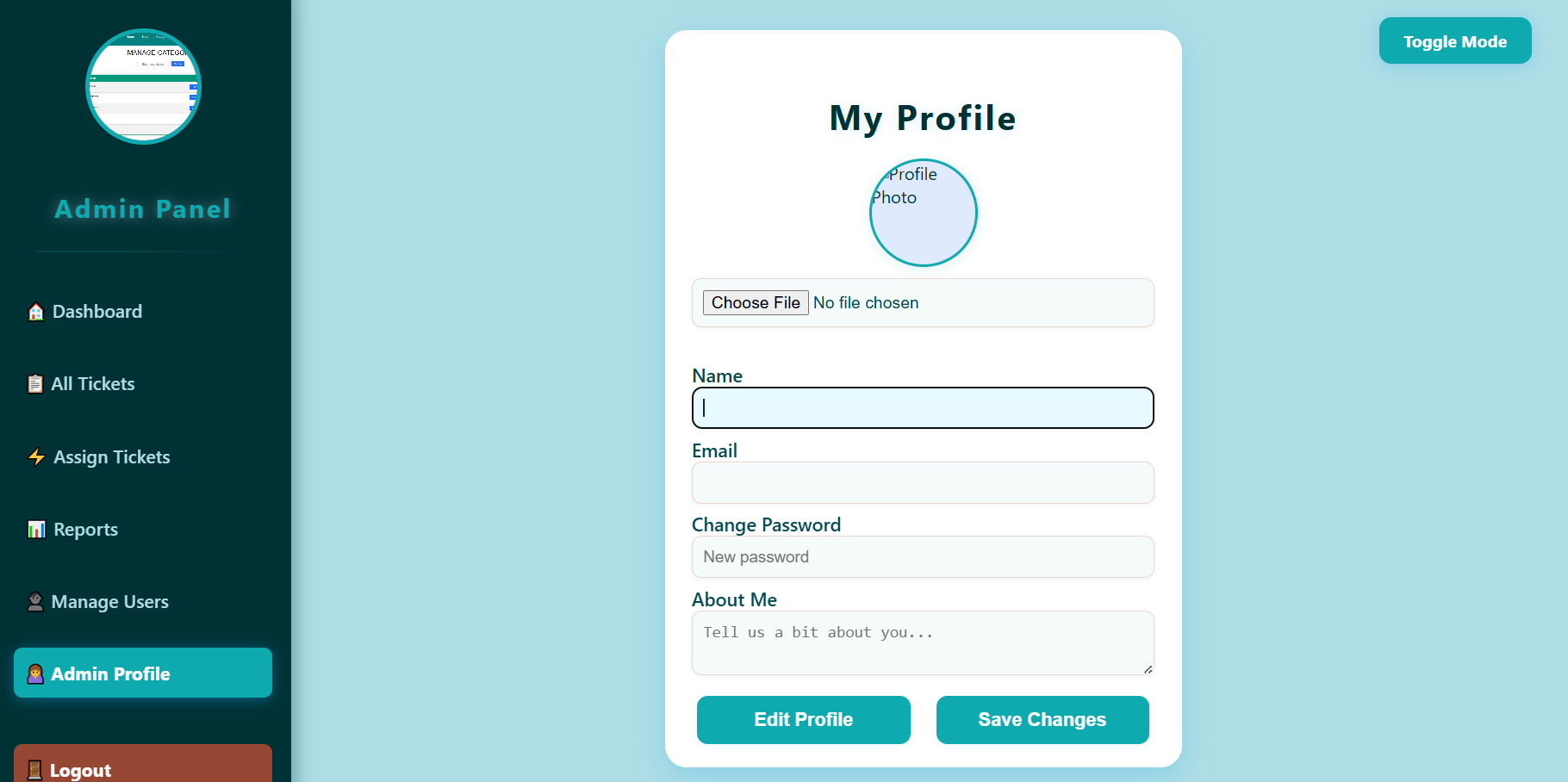
**Admin Panel**

Dark mode Light mode

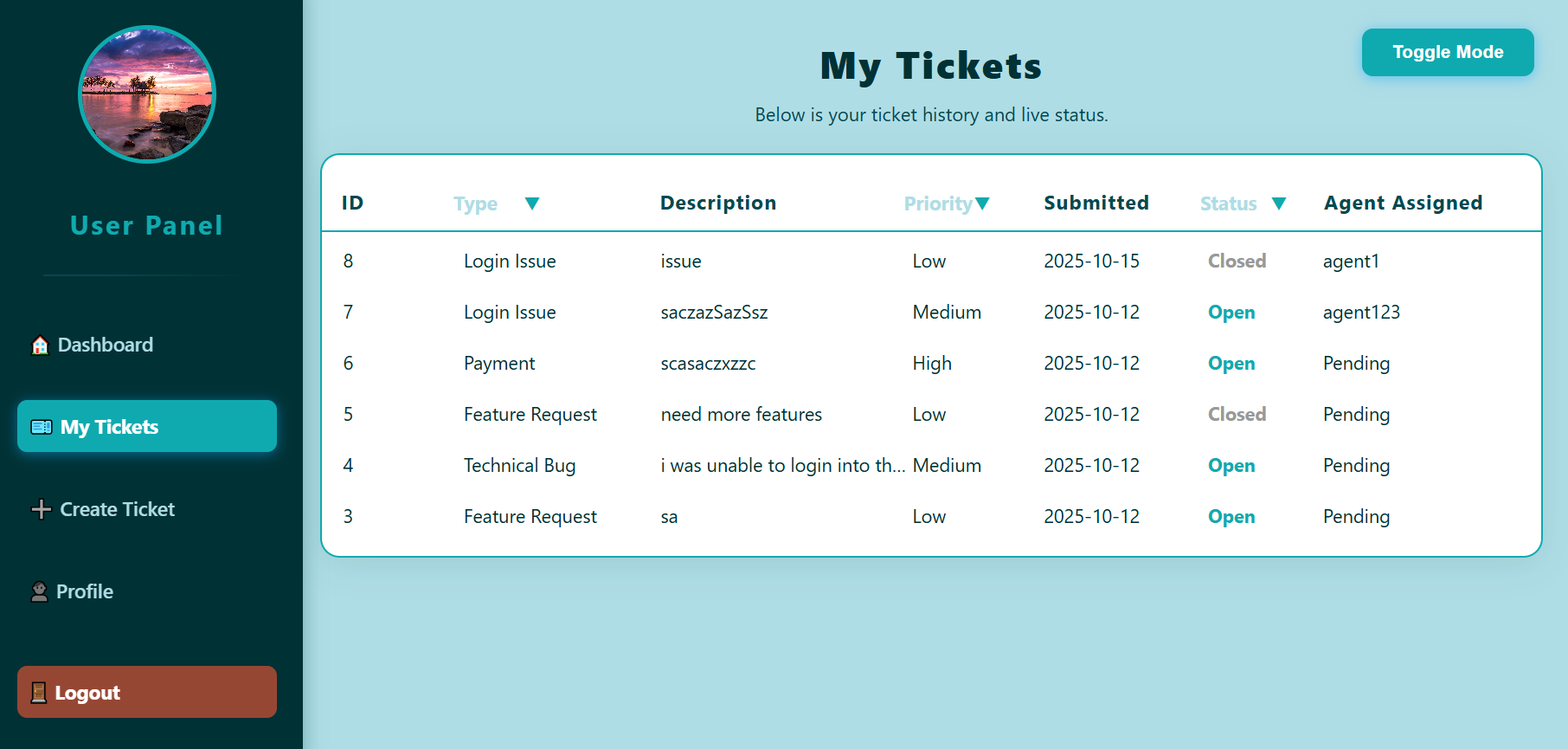
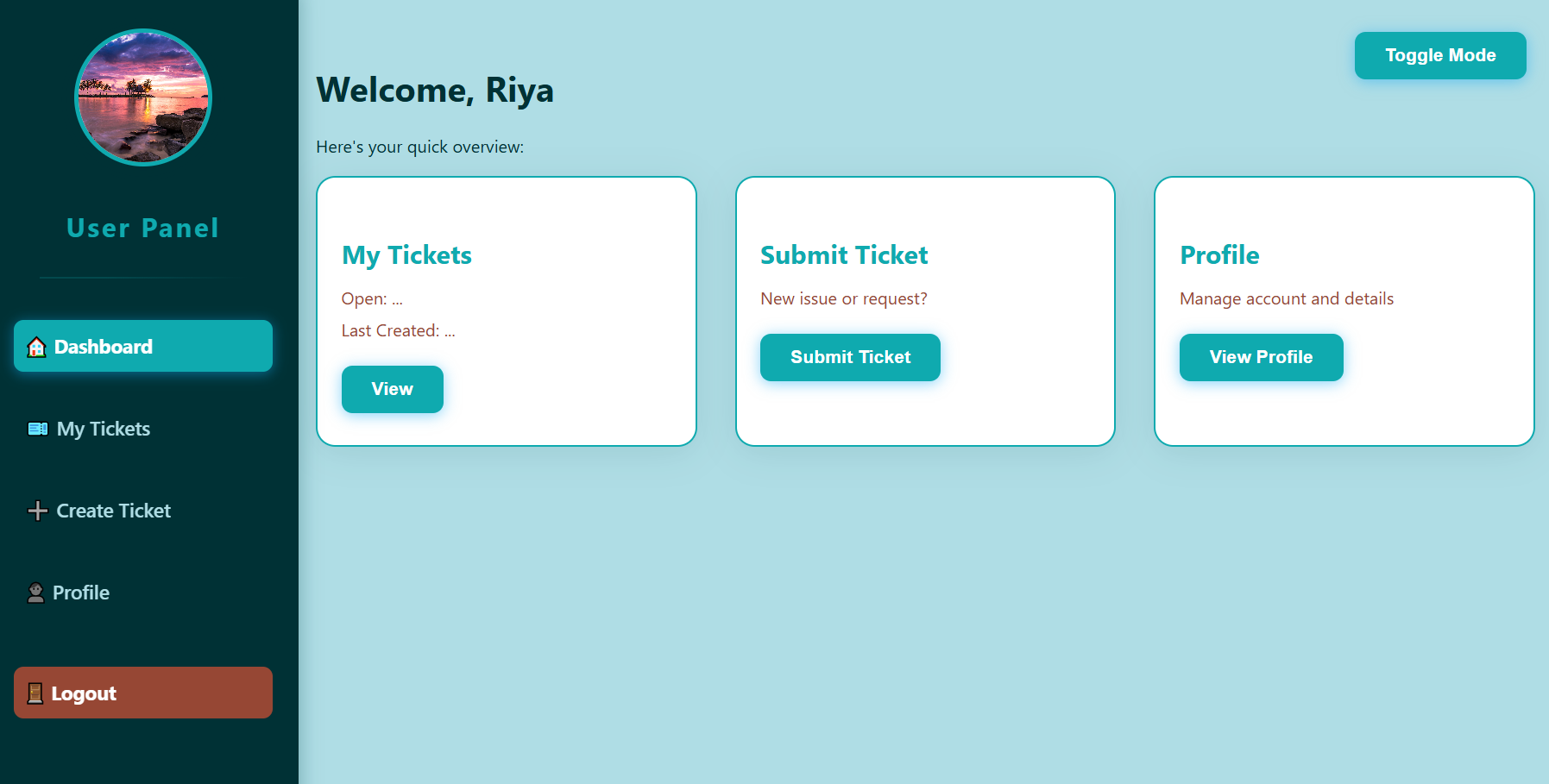
 

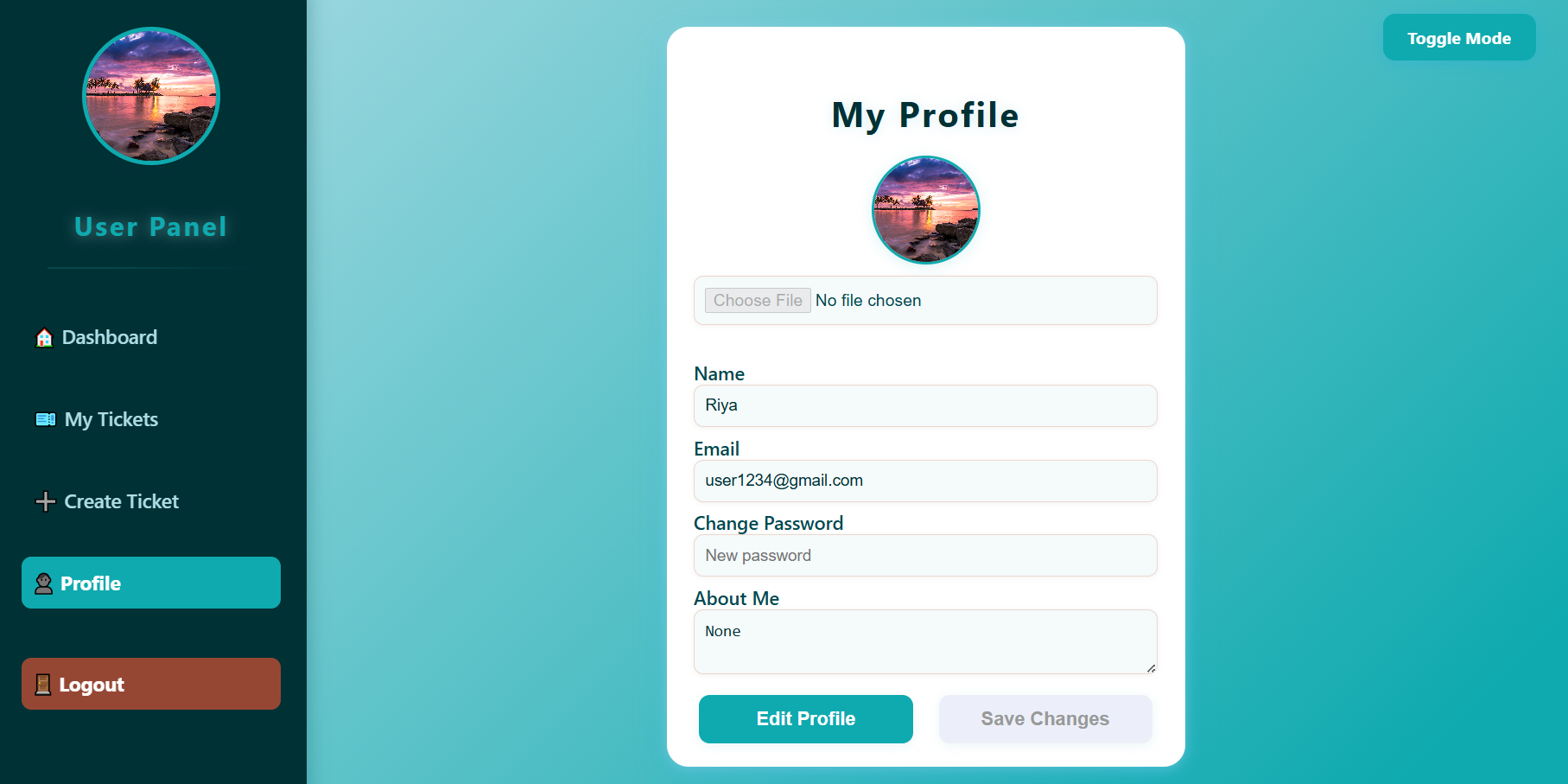
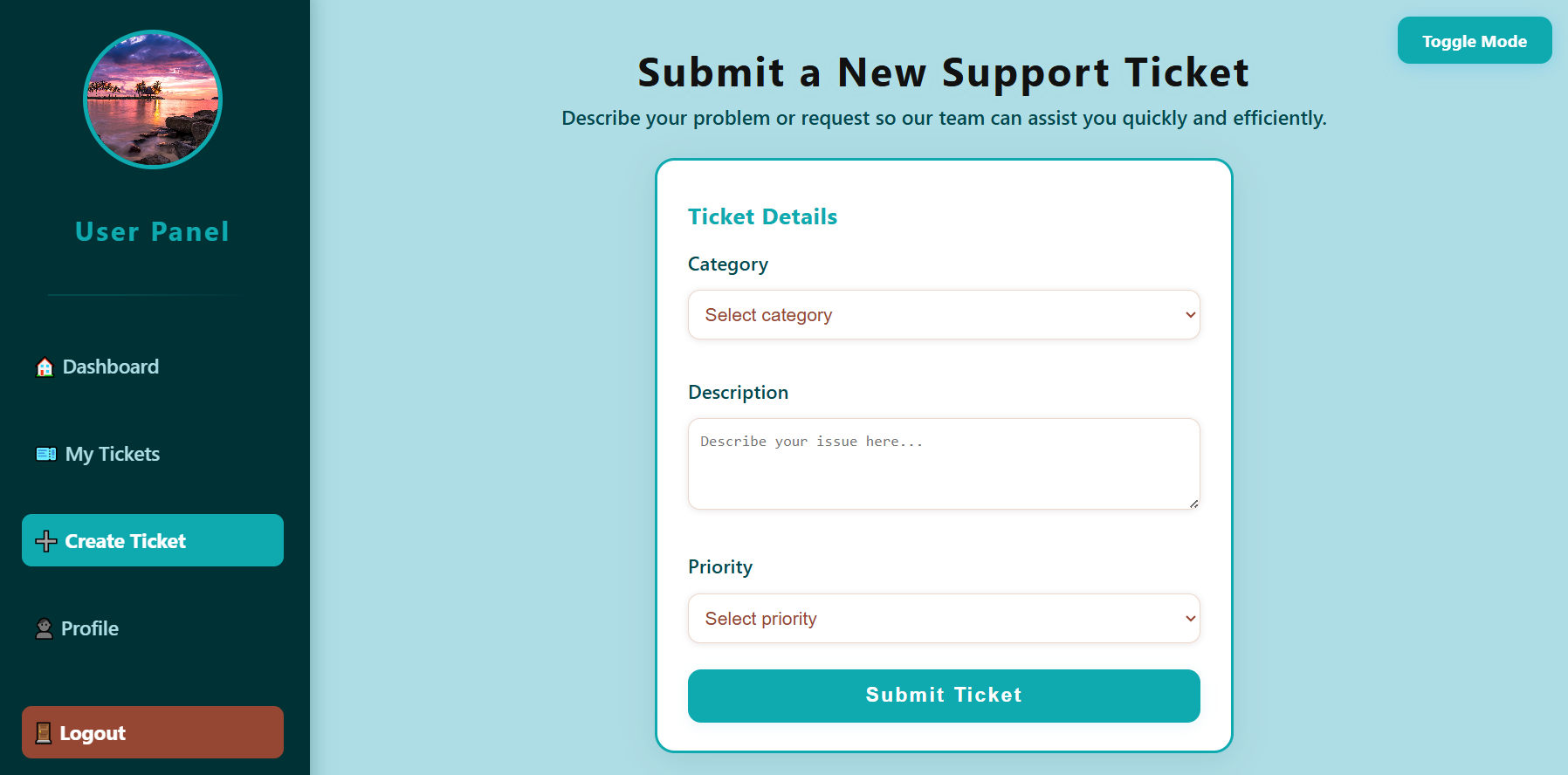
 



**User panel**





# Known Issues / Limitations

* No email notifications yet
* Basic role management (Admin, Agent, User)
* Limited file upload (profile photo only) - UI responsiveness can be improved

# Future Enhancements

* Add email notifications for updates
* Role-based dashboards for agents
* AI-driven ticket categorization
* File attachments in tickets
* SLA tracking and alerts

# Contact

Name: Zuveriya Tabassum Shaik

Email: tabassumzuveriya@gmail.com

GitHub: github.com/Zuveriya-Tabassum/support-ticket