T.C.

Marmara University Faculty of Engineering

Computer Engineering Department

CSE3055 Database Systems

Project Step #4 Report

Ahmet Sinan Kalkan 150119747

Ayşe Sena Aydemir 150119735  
Fatma Melisa Küçük 150119916  
Fatih Erkam Dinler 150119567

## **1. Project Description**

This project involves the development of a dormitory management system for a dormitory. The system is designed to manage student records, room assignments, maintenance and cleaning schedules, and billing processes. The application is built using MSSQL as the database and Node.js for the server-side logic, providing a comprehensive web-based interface for data management.

## **2. Scope**

The scope of this project covers the essential functionalities needed for dormitory management. Users can view and manage data, add new records, and update existing ones through the web interface. The project is built using **Node.js** for server-side logic, **EJS (Embedded JavaScript)** as the templating engine for rendering dynamic views, and **MSSQL** as the relational database system for storing and managing data. Additionally, the system leverages **stored procedures** and **triggers** in the database to automate key operations.

## **3. Project Run Guide**

This section explains how to set up, configure, and run the dormitory management system project. Follow these steps to ensure the system runs smoothly on your local machine.

Before starting, ensure you have the following installed:

* **Node.js** (LTS version recommended)
* **MSSQL Server** (Ensure the database engine is running)
* **npm** (Node Package Manager, installed with Node.js)

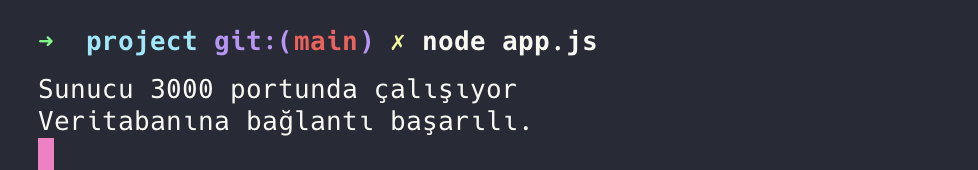
To start with, you should navigate the project folder and run the following command:

***npm install***

After that, you’ll need to set the connection string for the project. To do that, go the db.js file in the root of the project, and set the connection string as the following pattern:

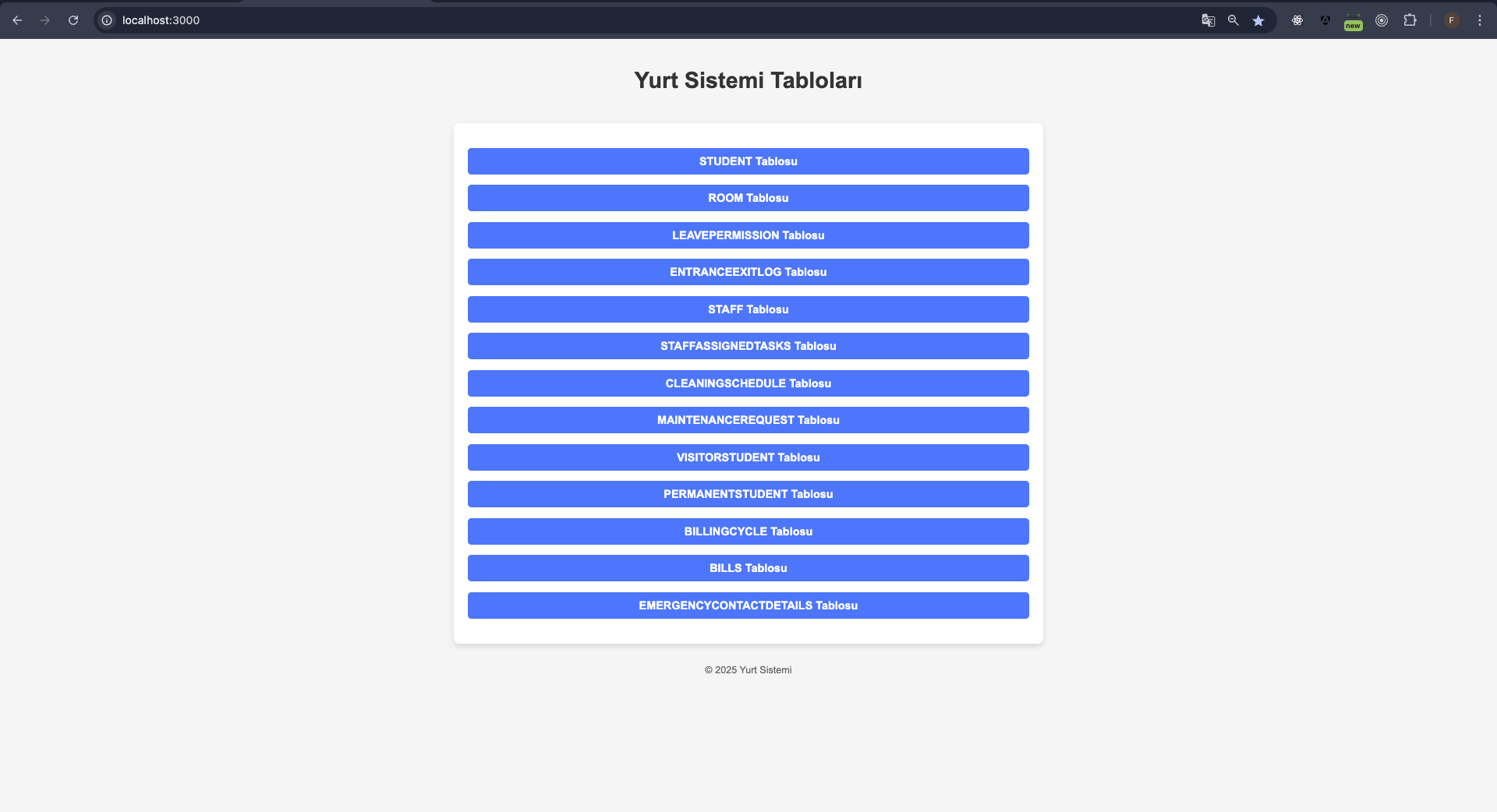


Then, type ``node app.js`` in the root of the project. This will create a connection to database and start a UI in the port 3000.



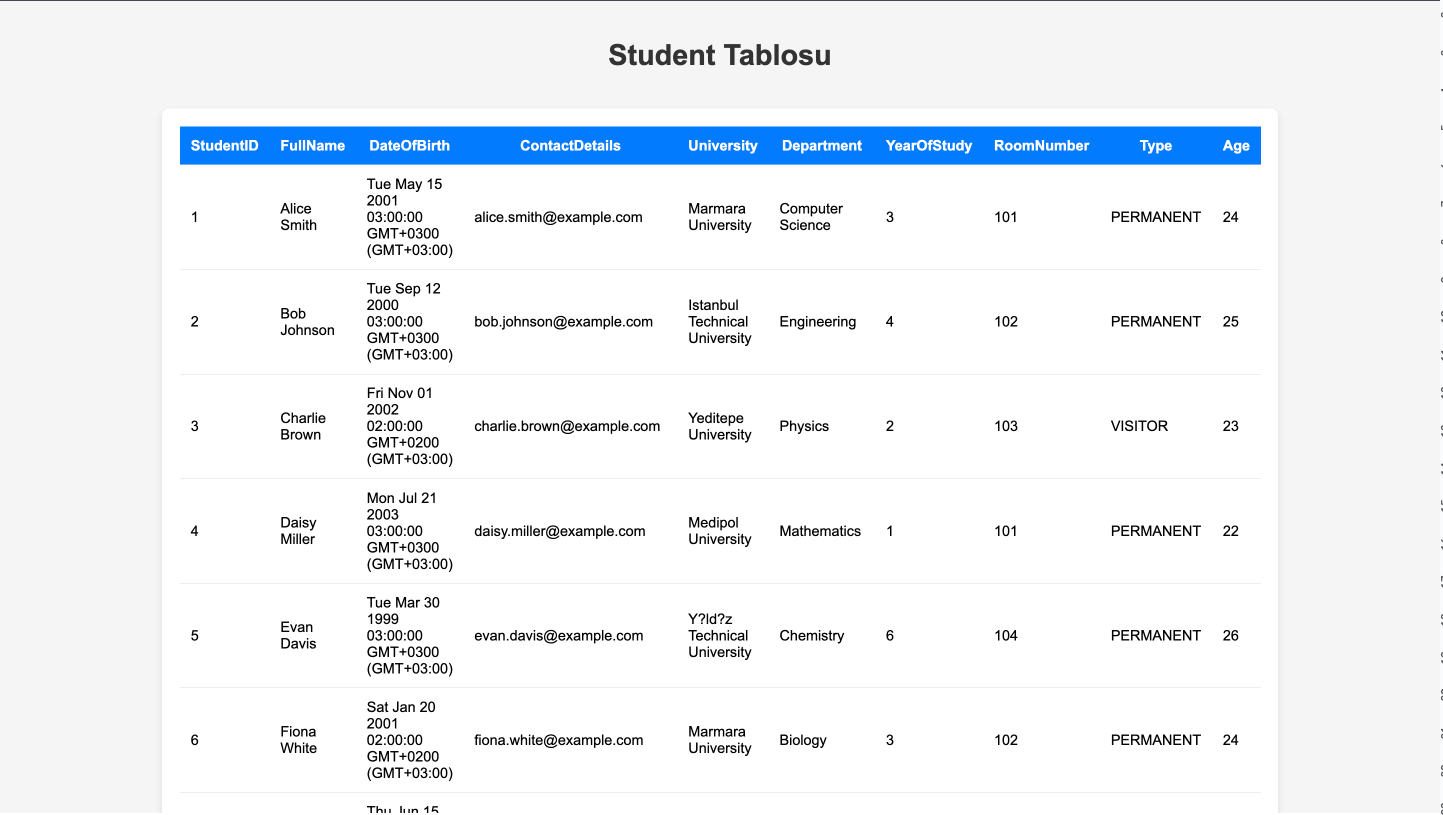
To see the web application in your local machine, open a browser and go to the following route:

``<http://localhost:3000/>``

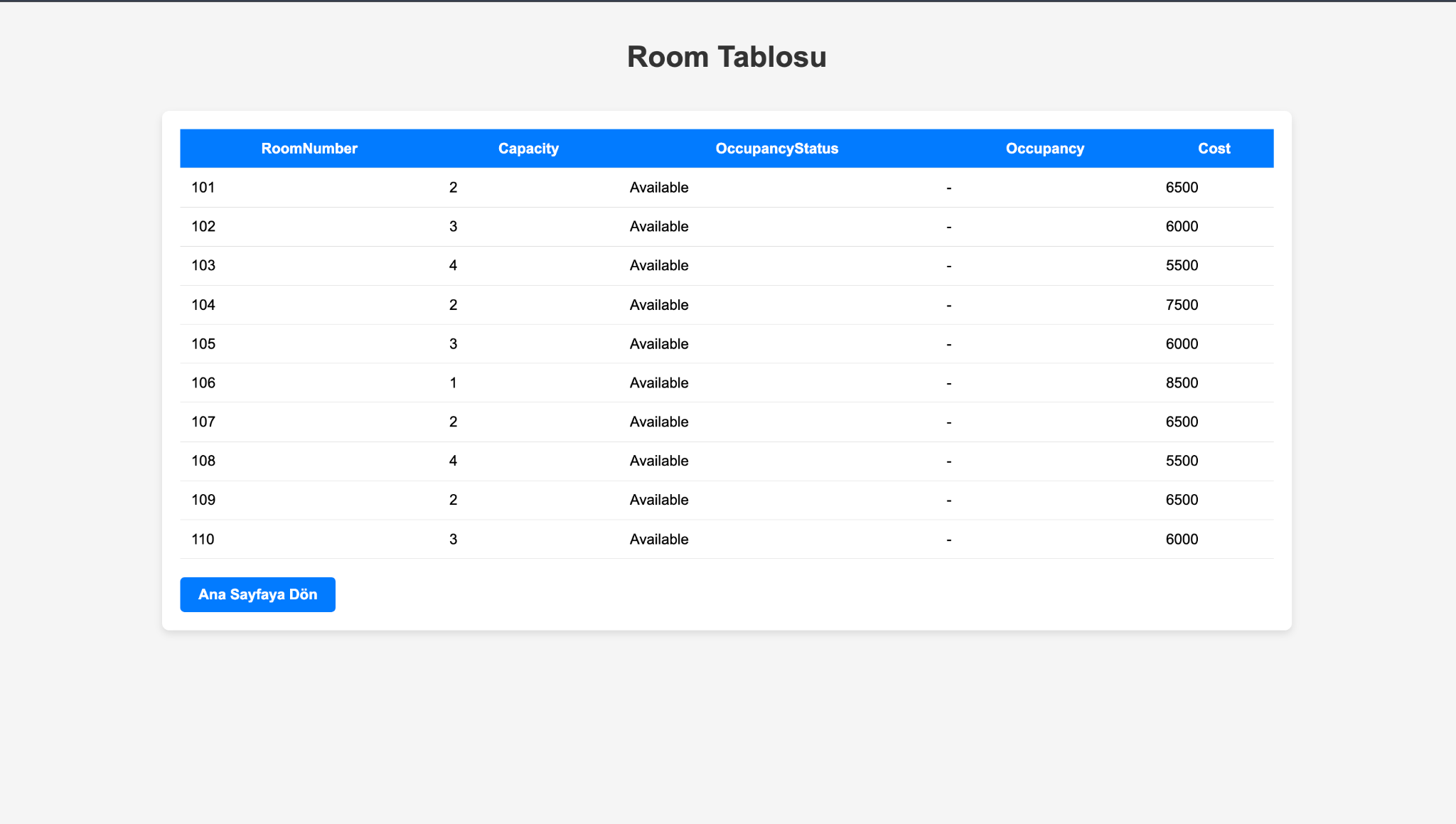
The dashboard page will be as follows:   


**4. Pages**

#### **4.1 Student Page**

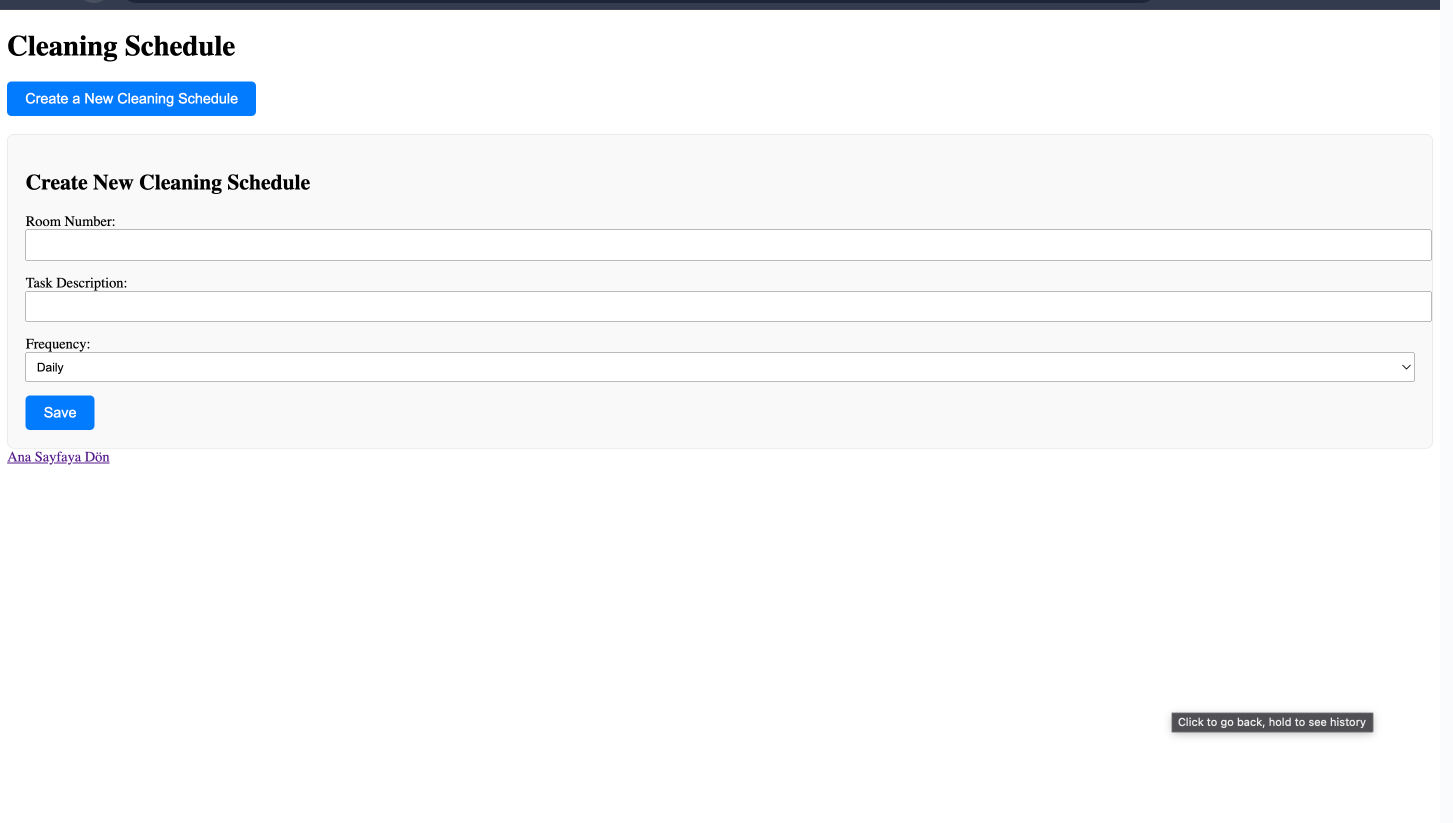
* **Purpose:** To display and manage student records.
* **Features:** A table displaying student details and a form for adding new student entries.
* 

#### **4.2 Room Page**

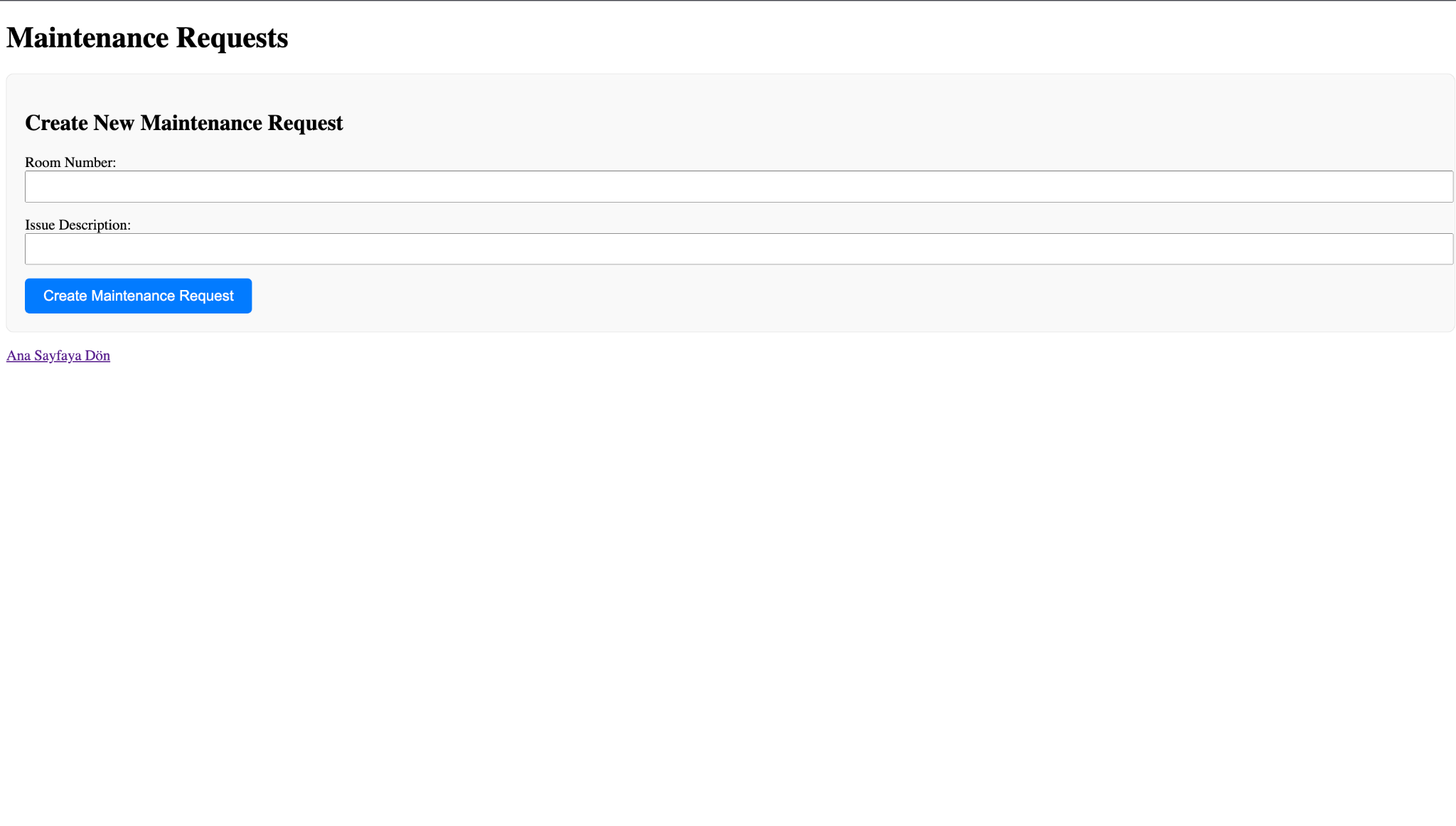
* **Purpose:** To display room details and occupancy rates.
* **Features:** A table listing all rooms with their statuses.
* 

#### 

#### **4.3 Cleaning Schedule Page**

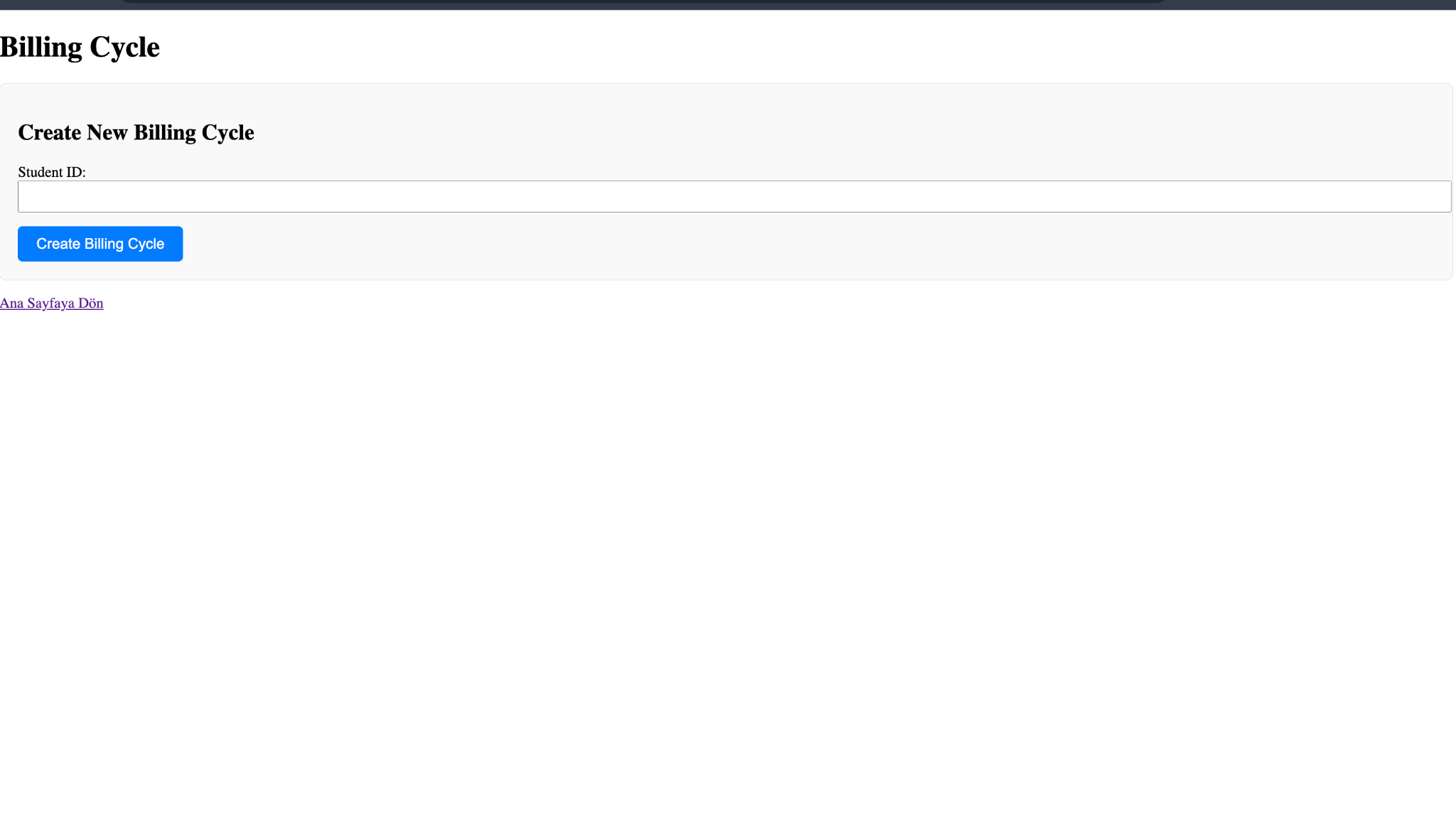
* **Purpose:** To plan and manage cleaning schedules.
* **Features:** A form for adding new cleaning schedules and a table displaying current schedules.
* 

#### **4.4 Maintenance Request Page**

* **Purpose:** To create and manage maintenance requests.
* **Features:** A form for submitting maintenance requests and a table showing ongoing requests.
* 

#### 

#### **4.5 Billing Cycle Page**

* **Purpose:** To create and manage student billing cycles and invoices.
* **Features:** A form for creating new billing cycles using a stored procedure.
* 

#### 

#### **4.6 Staff Page**

* **Purpose:** To manage dormitory staff details and roles.
* **Features:** A table displaying staff information, roles, and assigned tasks.
* 