**Unit: Internet Application Programming**

**Unit Code: ICS 2203**

1. RegNo **: SCT211-0012/2023** Name**: Ethan Kiptoo Kisang**
2. RegNo : **SCT211-0031/2023**  Name :**Koech Magdalene Chebet**

## Django Project Documentation

**Restaurant Reservation System**

### Project Overview

Provide a brief introduction to the project:

* **Name**: Restaurant Reservation System
* **Purpose**: To manage table reservations for a restaurant, with functionalities to add, edit and delete reservations.
* **Features**:
  + User can easily book a reservation of a flexible time and size of the party.
  + Admins can manage reservations and availability.

### Technologies Used

List the main technologies and frameworks:

* **Backend**: Django (Python)
* **Frontend**: HTML to make templates, CSS, JavaScript
* **Database**: SQLite
* **Others**: Bootstrap

### Distinctiveness and Complexity:

The project is unique, involving user role specialization with the admin acting as the superuser.

It has two applications, reservations and reserver as the base application.

**Complexity**

1. Dynamic Views: The web application involves creating, retrieving, updating, and deleting reservations with proper routing, forms, and templates.
2. Integration of Frontend and Backend: The system seamlessly integrates Bootstrap for styling and JavaScript for interactivity.

### Design Approach

1. **Model-View-Template (MVT) Architecture:**
   1. Models represent data structure.
   2. Views handle logic and retrieve data from the database.
   3. Templates render the data for display in a user-friendly format.
2. **Frontend Design:**
   1. Bootstrap Framework: Used for a clean and responsive UI.
3. **Routing:**
   1. Clear URL structure for creating, editing, viewing, and deleting reservations.

### File Structure and Descriptions

**Project Main Directory: reserver**

1. **settings.py**: Configuration file for the Django project, including database settings and app registration.
2. **urls.py:** Root URL configurations for the project.

**App Directory: reservations**

1. **models.py:** Contains the Reservation model, defining fields like customer\_name, date, time, and party\_size.
2. **views.py**: Implements logic for listing, creating, editing, and deleting reservations**.**
3. **forms.py**: Includes the ReservationForm class to manage form data.
4. **urls.py:** Maps specific views to their URLs.
5. **Templates directory:**
   * base.html: Base template with common layout and styles.
   * reservation\_list.html: Template for displaying a list of reservations.
   * reservation\_form.html: Template for creating and editing reservations.

**Static Files**

* **css:** Contains custom styles for the website.
* **js:** Includes JavaScript files for enhanced functionality, such as SweetAlert2 dialogs.

### How to Run the Application

1. **Set up the Environment:**
   * Install Python and Django.
2. **Install Dependencies:**

pip install django

1. **Make and update the Database:**

python manage.py makemigrations

python manage.py migrate

1. **Run the Server:**

python manage.py runserver

1. **Access the Application:**

Open a browser and navigate to http://127.0.0.1:8000/.

### Additional Information

**Improved User Experience:**

* + Bootstrap ensures a responsive and visually appealing interface.

**Potential Enhancements:**

* + Add user authentication for managing reservations per user.
  + Include features like reservation search and filtering.
  + Extend the database to manage restaurant tables and availability.