

Doctor Management System Report

1. Project Overview

This project is a real-time Doctor Management System built in Python. It allows adding, viewing, searching, updating, and deleting doctor records. Data is stored in a JSON file, making the system persistent even after restart. This version includes Email and Contact fields, sorting, and search features.

2. Technologies Used

- Python
- JSON file storage
- os module for file existence check

3. System Features

- Add doctor with name, specialization, email, and contact
- Prevents duplicate doctor names
- Auto-generates unique ID for each doctor
- Display doctors with sorting options (id/name/specialization)
- Search doctors by name or specialization
- Update specialization, email, and contact individually
- Delete doctor with confirmation
- Real-time persistent data storage

4. Class Breakdown

A) Doctor Class:

- Stores individual doctor information

- **Attributes:** id, name, specialization, email, contact
- **Method:** to_dict() converts doctor data to dictionary format

B) DoctorDatabase Class:

- Handles data storage and retrieval
- Uses JSON file 'doctors.json'
- Methods include:
 - _load() – loads data from JSON
 - _save() – writes data to JSON
 - generate_id() – creates new unique ID
 - add_doctor() – adds new doctor
 - get_all_doctors() – returns doctor list with sorting
 - search_doctors() – searches by name or specialization
 - get_doctor_by_id() – finds doctor using ID
 - update_doctor() – updates selected doctor fields
 - delete_doctor() – removes doctor from system

C) DoctorSystem Class:

- Provides user interface through menu
- Connects user inputs with database functions
- Handles:
 - Add

- Show
- Search
- Update
- Delete
- Exit

5. Data Storage Structure (JSON Format)

```
{
```

```
    "last_id": 3,
```

```
    "doctors": [
```

```
        {
```

```
            "id": 1,
```

```
            "name": "Dr. Ali",
```

```
            "specialization": "Cardiologist",
```

```
            "email": "ali@example.com",
```

```
            "contact": "03001234567"
```

```
        }
```

```
    ]
```

```
}
```

6. Sample System Output

```
===== REAL-TIME DOCTOR MANAGEMENT =====
```

1. Add Doctor
2. Show Doctors
3. Search Doctors

4. Update Doctor

5. Delete Doctor

6. Exit

----- Doctor List -----

ID	Name	Specialization	Email	Contact
1	Dr. Ali	Cardiologist	ali@example.com	03001234567

7. Limitations

- No file locking (risk of overwrite in multi-user use)
- No login or authentication
- No data validation for email format or number format

8. Future Improvements

- Add authentication (admin login)
- Add file locking for safety
- Getting valid email format
- Avoid duplicates in contacts