 CONTACT MANAGEMENT SYSTEM

**A PROJECT REPORT**

***Submitted by***

**OM PRAKASH M (2303811724321079)**

***in partial fulfillment of requirements for the award of the course***

**CGB1221-DATABASE MANAGEMENT SYSTEMS**

***In***

### ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

# SAMAYAPURAM – 621 112

**JUNE- 2025**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

**(AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

# BONAFIDE CERTIFICATE

Certified that this project report on **“CONTACT MANAGEMENT SYSTEM”** is the bonafide work of **OM PRAKASH M (2303811724321079)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  Dr.T.AVUDAIAPPAN, M.E.,Ph.D.,  **HEAD OF THE DEPARTMENT**  ASSOCIATE PROFESSOR  Department of Artificial Intelligence  K.Ramakrishnan College of Technology (Autonomous)  Samayapuram–621112. | **SIGNATURE**  Mrs.S.GEETHA, M.E.,  **SUPERVISOR**  ASSISTANT PROFESSOR  Department of Artificial Intelligence  K.Ramakrishnan College of Technology (Autonomous)  Samayapuram–621112. |

Submitted for the viva-voce examination held on …04.06.2025……………….

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**DECLARATION**

I declare that the project report on **“CONTACT MANAGEMENT SYSTEM”** is the result of original work done by me and best of my knowledge, similar workhas not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1221 – DATABASE MANAGEMENT SYSTEMS.**

.

**Signature**

OM PRAKASH M

Place: Samayapuram

Date:04.06.2025

**ACKNOWLEDGEMENT**

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequateduration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr.S.GEETHA, M.E.,** Department of **ARTIFICIAL INTELLIGENCE,** for his incalculable suggestions, creativity, assistance and patiencewhich motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

**INSTITUTE**

**Vision:**

* To serve the society by offering top-notch technical education on par with global standards.

**Mission:**

* Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
* Be an institute with world class research facilities.
* Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

**DEPARTMENT**

**Vision:**

* To excel in education, innovation, and research in Artificial Intelligence and Data Science to fulfil industrial demands and societal expectations.

**Mission**

* To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.
* To collaborate with industry and offer top-notch facilities in a conducive learning environment.
* To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.
* To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

**PROGRAM EDUCATIONAL OBJECTIVES (PEO)**

* **PEO1:** Compete on a global scale for a professional career in Artificial Intelligence and Data Science.
* **PEO2:** Provide industry-specific solutions for the society with effective communication and ethics.
* **PEO3** Enhance their professional skills through research and lifelong learning initiatives.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

* **PSO1:** Capable of finding the important factors in large datasets, simplify the data, and improve predictive model accuracy.
* **PSO2:** Capable of analyzing and providing a solution to a given real-world problem by designing an effective program.

**PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**ABSTRACT**

The Contact Management System is a database-driven application designed to efficiently store, manage, and retrieve personal or professional contact information. The system allows users to perform core database operations such as adding, viewing, updating, and deleting contact records. Each contact entry typically includes details such as name, phone number, email address, and physical address. Developed using a relational database management system (RDBMS**)**, the project emphasizes the practical application of database design principles, including normalization, entity-relationship modeling, and query execution through SQL. By implementing this system, users can experience a simplified and organized method of managing contact data, reducing redundancy and ensuring data consistency. This project serves as a foundational exercise in understanding how databases can support real-world applications. It demonstrates the effective use of structured query language (SQL) and relational data models to handle dynamic datasets, reinforcing key concepts in database management systems.

# 

# ABSTRACT WITH POs AND PSOs MAPPING

# CO 5 : BUILD DATABASES FOR SOLVING REAL-TIME PROBLEMS

| **ABSTRACT** | **POs MAPPED** | **PSOs MAPPED** |
| --- | --- | --- |
| This project presents a Contact Management System developed using Flask (Python web framework) and SQLite as the backend database. The system allows users to efficiently manage personal or professional contacts through a user-friendly interface. Core functionalities include creating, reading, updating, and deleting contact entries (CRUD operations). Each contact can include details such as name, phone number, email, address, and group categorization. | **PO1 -3**  **PO2 -2**  **PO3 -3**  **PO4 -2**  **PO5 -3**  **PO6 -1**  **PO7 -1**  **PO8 -2**  **PO9 -2**  **PO10 -2**  **PO11-2**  **PO12 -3** | **PSO1 -3**  **PSO2 -2** |

Note: 1- Low, 2-Medium, 3- High

**TABLE OF CONTENTS**

| **CHAPTER**  **NO.** | TITLE | **PAGE**  **NO.** |
| --- | --- | --- |
|  | ABSTRACT |  |
| 1 | INTRODUCTION | 1 |
|  | 1.1 OBJECTIVE | 1 |
|  | 1.2 OVERVIEW | 2 |
|  | 1.3 SQL AND DATABASE CONCEPTS | 3 |
| 2 | PROJECT METHODOLOGY | 4 |
|  | 2.1 PROPOSED WORK | 5 |
|  | 2.2 BLOCK DIAGRAM | 6 |
| 3 | MODULE DESCRIPTION | 6 |
|  | 3.1 3.1 USER INTERFACE MODULE | 7 |
|  | 3.2 DATABASE MANAGEMENT MODULE | 7 |
|  | 3.3 SEARCH & SORTING MODULE | 7 |
|  | 3.4 IMPORT & EXPORT MODULE | 8 |
|  | 3.5 ADMIN MODULE | 8 |
| 4 | CONCLUSION & FUTURE SCOPE | 9 |
|  | APPENDIX A SOURCE CODE | 10 |
|  | APPENDIX B SCREENSHOTS | 13 |
|  | REFERENCES | 15 |
|  |  |  |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION**

The “Contact Management System” is a software solution designed to efficiently store and manage contact information. It allows users to add, edit, delete, search, sort, import, and export contacts easily. This system replaces traditional manual methods with a centralized digital platform. Users can securely manage names, phone numbers, emails, and other personal details. The interface is designed to be simple, intuitive, and user-friendly. Search and sorting features help users find contacts quickly and stay organized. Import functionality allows bulk uploading of contacts from various file formats. Export options enable users to back up or share contact data when needed.

* 1. **OBJECTIVE**

The main objectives of the Contact Management System Using DBMS project are:

* To develop a centralized system for storing and managing contact information.
* To provide an easy-to-use interface for adding, editing, viewing, and deleting contacts.
* To implement efficient search and sorting functionalities for quick data access.
* To enable users to import contacts in bulk from external files like CSV or Excel.
* To allow export of contact data in multiple formats for backup and sharing.
* To support organized contact grouping and categorization (if implemented).
* To maintain data accuracy and consistency through database management.

**1.2 OVERVIEW**

The Contact Management System is a database-driven application designed to streamline the process of storing and managing contact information. It offers a centralized platform where users can efficiently add, view, edit, and delete contact details such as names, phone numbers, emails, and addresses. The system features a user-friendly interface along with powerful search and sorting capabilities, allowing users to quickly find and organize their contacts. Additionally, it supports importing contact data from external sources like CSV or Excel files, as well as exporting contacts for backup or sharing purposes. An admin module is integrated to manage user roles and maintain system performance. This project aims to enhance productivity, reduce manual errors, and provide a reliable solution for both personal and professional contact management needs.

The system supports the following key features:

* **User-friendly interface** (optional: if a frontend is developed)
* **Secure and reliable data storage**
* **CRUD operations** (Create, Read, Update, Delete)
* **Search and filter functionality** for quick access to specific contacts
* **Scalable structure** allowing future enhancements like login systems, contact grouping, and import/export options

This project serves as a learning platform to apply DBMS concepts in practice. It provides insight into designing and interacting with relational databases, writing efficient SQL queries, and understanding how backend systems support data-driven applications. The Contact Management System can also be extended into a more robust application with a graphical user interface or integrated into larger customer relationship management (CRM) system.

* 1. **SQL AND DATABASE CONCEPT:**

A database is an organized collection of structured data that can be easily accessed, managed, and updated. In a Contact Management System, the database stores user and contact details securely and efficiently.

**Key Database Concepts:**

* **Tables**:  
  Data is stored in tabular form (rows and columns). Each table represents an entity like “Users” or “Contacts”.
* **Primary Key**:  
  A unique identifier for each record in a table (e.g., contact\_id in the Contacts table).
* **Foreign Key**:  
  A field that links two tables together. It maintains relationships (e.g., user\_id in Contacts linking to Users table).
* **Normalization**:  
  Process of organizing data to reduce redundancy and improve data integrity.

**SQL (Structured Query Language)**

SQL is the standard language used to communicate with databases. It helps in performing operations like inserting, retrieving, updating, and deleting data.

**Key SQL Concepts:**

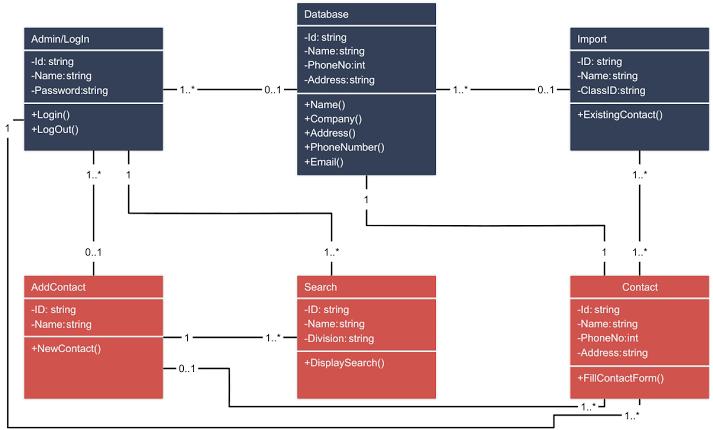
* **Data Definition Language (DDL)**:  
  Used to define and modify the database structure.
  + CREATE TABLE, ALTER TABLE, DROP TABLE
* **Data Manipulation Language (DML)**:  
  Used to handle data within the tables.
  + INSERT INTO, SELECT, UPDATE, DELETE
* **Data Control Language (DCL)**:  
  Used to control access to data.
* **Queries**:  
  Instructions written in SQL to interact with data (e.g., searching contacts by name).
* **Joins**:  
  Combines rows from two or more tables based on related columns

**2.PROJECT METHODOLOGY**

**2.1 PROPOSED SYSTEM**

The proposed work for this project involves the following specific tasks and deliverables:

1. **Requirement Gathering and Analysis**  
   Identify functional and non-functional requirements, user roles, and feature scope for the system.
2. **System and Database Design**  
   Design the system architecture, user interface wireframes, and relational database schema with entity relationships.
3. **Frontend and Backend Development**  
   Develop a responsive user interface and implement backend logic to manage user actions and database interaction.
4. **Module Implementation**  
   Build and integrate core modules including Add/Edit Contacts, Search & Sort, Import, Export, and Admin Control.
5. **Database Setup and Integration**  
   Configure the database, create required tables, and ensure secure and efficient data operations using SQL.
   1. **BLOCK DIAGRAM**



**CHAPTER 3**

**MODULE DESCRIPTION**

**3.1 USER INTERFACE MODULE**

Provides a user-friendly interface for adding, editing, deleting, and viewing contacts.It ensures smooth navigation and real-time feedback for efficient interaction.

**Key Features:**

* Clean and intuitive UI for adding, updating, viewing, and deleting contacts.
* Responsive design for desktop and mobile access.

**3.2 DATABASE MANAGEMENT MODULE**

Handles all contact data storage, retrieval, and update operations using SQL.Ensures data consistency, integrity, and optimized performance.  
**Key Features:**

* Stores and organizes contact information using a relational database (e.g., SQLite, MySQL).
* Executes SQL queries for CRUD (Create, Read, Update, Delete) operations.

**3.3 SEARCH AND SORTING MODULE**

Allows users to quickly find contacts using name, phone, or email filters.  
Supports sorting by various criteria for organized data presentation**.  
Key Features:**

* Stores and organizes contact information using a relational database (e.g., SQLite, MySQL).
* Executes SQL queries for CRUD (Create, Read, Update, Delete) operations.

**3.4 IMPORT MODULE**

Enables users to upload bulk contacts from CSV, Excel, or JSON files.  
Validates data formats and provides feedback on import errors.  
**Key Features:**

* Supports importing contacts from CSV, Excel, or JSON files.
* Pre-import validation to check format, duplicates, and required fields.

**3.5 EXPORT MODULE**

Lets users download contact data in preferred formats like CSV or Excel.  
Useful for creating backups or sharing contact lists efficiently.

**Key Features:**

* Exports full or filtered contact lists in formats such as CSV, Excel, or PDF.
* Option to select specific fields to include in the export.

**3.6 ADMIN MODULE**

Allows administrators to manage users, control access, and monitor system performance.Provides tools for database maintenance, logs, and configuration. **Key Features**:

* User management: add/remove users, assign roles.
* System configuration: manage permissions, view logs, backup database.

**CHAPTER 4**

**CONCLUSION & FUTURE SCOPE**

**CONCLUSION:**

The Contact Management System developed in this project successfully demonstrates the application of Database Management System (DBMS) concepts in a real-world scenario. By integrating Flask as the web framework and SQLite as the backend database, the system provides an efficient and user-friendly interface to perform all essential operations such as adding, viewing, updating, deleting, searching, and grouping contacts.

The project showcases the use of a relational database to store structured data, SQL queries for data manipulation and retrieval, and RESTful APIs for data communication between the frontend and backend. It also includes CSV import/export functionality to support data portability, which is crucial for data backup and migration.

From a DBMS perspective, the project highlights key concepts such as:

* Data modeling and table design
* Primary key constraints for unique identification
* Data integrity and consistency
* Query optimization using conditional filtering
* Secure and modular interaction with the database via Flask

**FUTURE SCOPE:**

While the current version of the Contact Management System meets the fundamental requirements of a database-driven contact application, there are several potential areas for enhancement and expansion in the future

**1.Authentication and User Management**

Implementing a login system with user authentication and role-based access control (e.g., admin vs. regular user) can improve security and allow multiple users to manage their own contact lists.

**2. Cloud Database Integration**

Migrating the database from SQLite to cloud-based solutions like MySQL, PostgreSQL, or Firebase will allow for better scalability and remote access.

**3.Mobile Application Support**

Developing a mobile-friendly version or integrating with a mobile app (using frameworks like React Native or Flutter) will enable on-the-go access and synchronization.

**4.Data Security and Backup**

Adding encryption for sensitive data (like emails or phone numbers) and automatic backup options will enhance data protection and reliability.

**5.Integration with External Services**

Features like syncing with Google Contacts, exporting to Excel, or sending emails directly from the app can be added for seamless interoperability.

**6.Notifications and Reminders**

Users could set reminders for contacting someone or receive notifications about updates or important contacts.

**APPENDIX A SOURCE CODE**

! from flask import Flask, request, jsonify, render\_template

import sqlite3

import csv

import io

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

# Establish connection to the SQLite database

def get\_db\_connection():

conn = sqlite3.connect('contacts.db') # File-based database

conn.row\_factory = sqlite3.Row # Enable row access by column name

return conn

# Initialize the database and create table if not exists

def init\_db():

conn = get\_db\_connection()

c = conn.cursor()

c.execute('''CREATE TABLE IF NOT EXISTS contacts

(id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

phone TEXT,

email TEXT,

address TEXT,

group\_name TEXT)''')

conn.commit()

conn.close()

# Initialize database on app start

init\_db()

# Route for the main page

@app.route('/')

def index():

return render\_template('index.html')

# Get all contacts or filter by search/group

@app.route('/contacts', methods=['GET'])

def get\_contacts():

search = request.args.get('search', '')

group = request.args.get('group', '')

conn = get\_db\_connection()

c = conn.cursor()

query = 'SELECT \* FROM contacts WHERE 1=1'

params = []

if search:

query += ' AND (name LIKE ? OR phone LIKE ? OR email LIKE ? OR address LIKE ?)'

params.extend([f'%{search}%'] \* 4)

if group:

query += ' AND group\_name = ?'

params.append(group)

c.execute(query, params)

contacts = [{'id': row['id'], 'name': row['name'], 'phone': row['phone'], 'email': row['email'],

'address': row['address'], 'group\_name': row['group\_name']} for row in c.fetchall()]

conn.close()

return jsonify(contacts)

# Add a new contact

@app.route('/contacts', methods=['POST'])

def add\_contact():

data = request.json

conn = get\_db\_connection()

c = conn.cursor()

c.execute('INSERT INTO contacts (name, phone, email, address, group\_name) VALUES (?, ?, ?, ?, ?)',

(data['name'], data['phone'], data['email'], data['address'], data['group\_name']))

conn.commit()

contact\_id = c.lastrowid

conn.close()

return jsonify({'id': contact\_id})

# Update an existing contact

@app.route('/contacts/<int:id>', methods=['PUT'])

def update\_contact(id):

data = request.json

conn = get\_db\_connection()

c = conn.cursor()

c.execute('UPDATE contacts SET name=?, phone=?, email=?, address=?, group\_name=? WHERE id=?',

(data['name'], data['phone'], data['email'], data['address'], data['group\_name'], id))

conn.commit()

conn.close()

return jsonify({'message': 'Contact updated'})

# Delete a contact

@app.route('/contacts/<int:id>', methods=['DELETE'])

def delete\_contact(id):

conn = get\_db\_connection()

c = conn.cursor()

c.execute('DELETE FROM contacts WHERE id=?', (id,))

conn.commit()

conn.close()

return jsonify({'message': 'Contact deleted'})

# Get all distinct groups

@app.route('/groups', methods=['GET'])

def get\_groups():

conn = get\_db\_connection()

c = conn.cursor()

c.execute('SELECT DISTINCT group\_name FROM contacts WHERE group\_name IS NOT NULL')

groups = [row['group\_name'] for row in c.fetchall()]

conn.close()

return jsonify(groups)

# Export all contacts to CSV

@app.route('/export', methods=['GET'])

def export\_contacts():

conn = get\_db\_connection()

c = conn.cursor()

c.execute('SELECT name, phone, email, address, group\_name FROM contacts')

output = io.StringIO()

writer = csv.writer(output)

writer.writerow(['Name', 'Phone', 'Email', 'Address', 'Group'])

writer.writerows(c.fetchall())

conn.close()

return jsonify({'csv': output.getvalue()})

# Import contacts from CSV

@app.route('/import', methods=['POST'])

def import\_contacts():

file = request.files['file']

stream = io.StringIO(file.stream.read().decode('UTF-8'))

reader = csv.reader(stream)

next(reader) # Skip header

conn = get\_db\_connection()

c = conn.cursor()

for row in reader:

c.execute('INSERT INTO contacts (name, phone, email, address, group\_name) VALUES (?, ?, ?, ?, ?)', row)

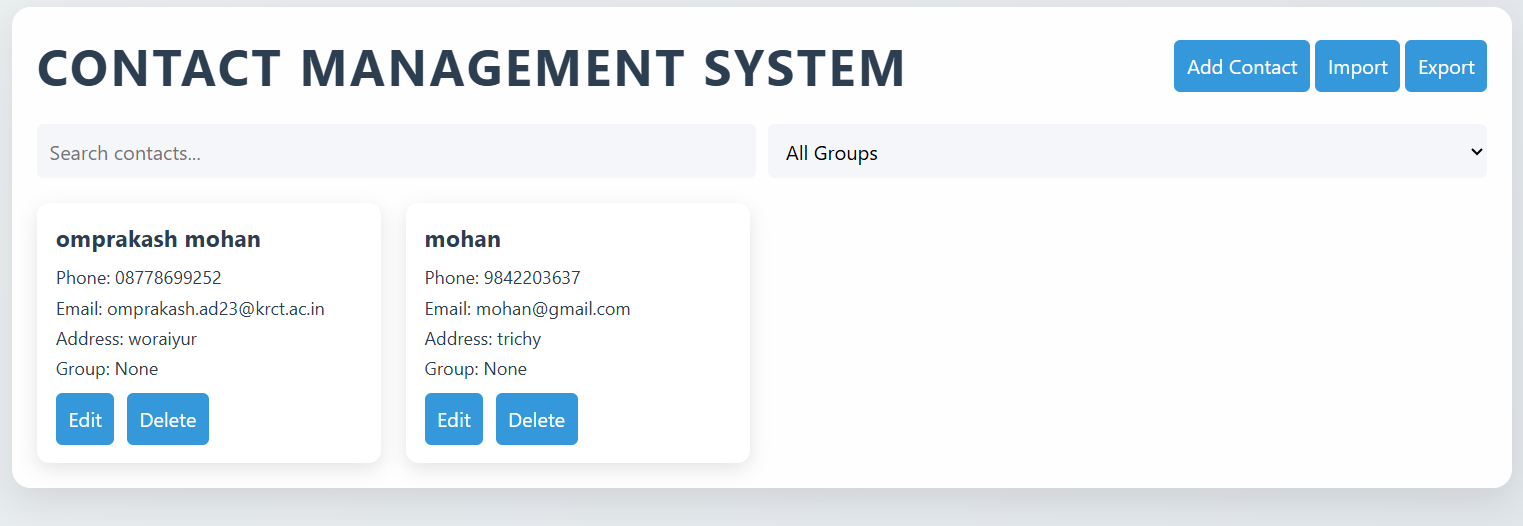
conn.commit()

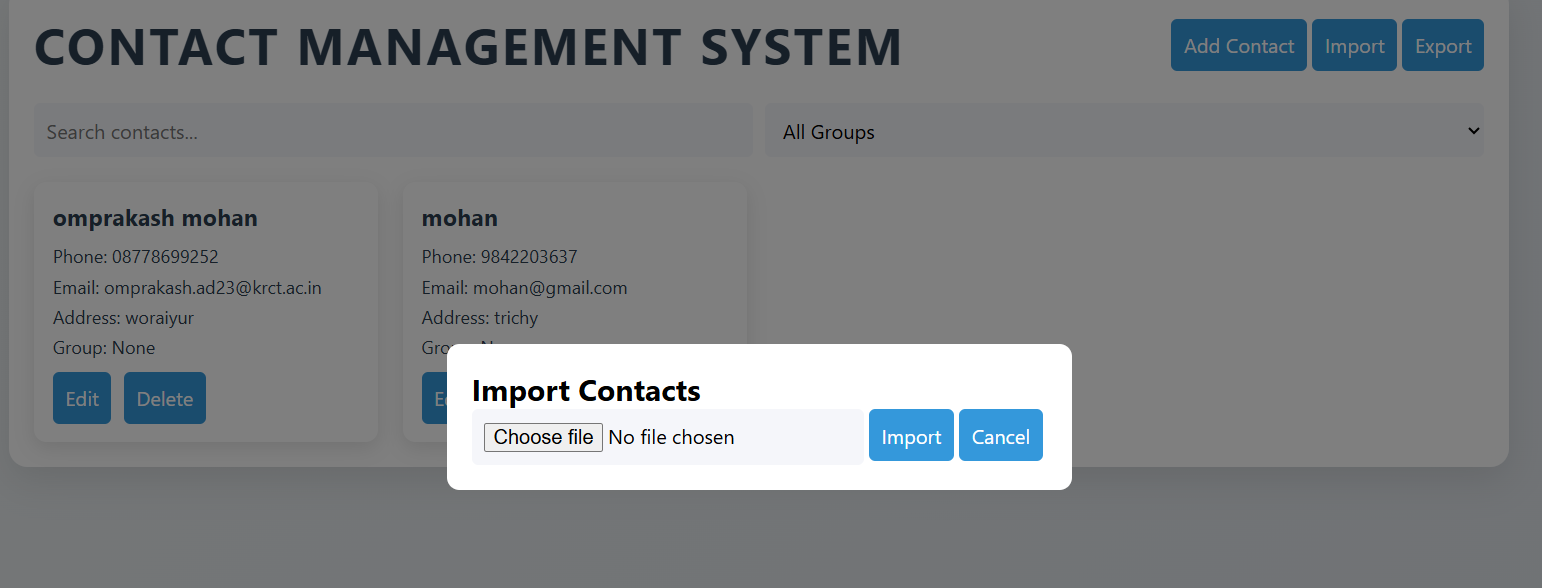
conn.close()

return jsonify({'message': 'Contacts imported'})

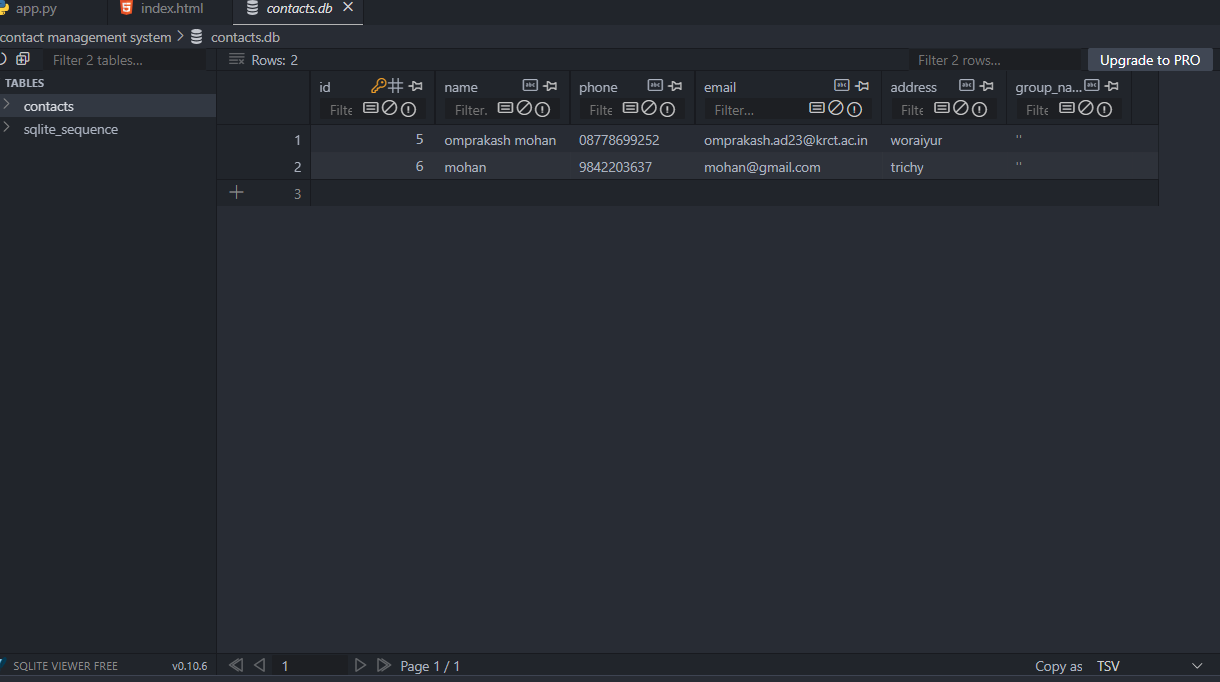
# Run the Flask app

**APPENDIX B SCREENSHOTS**

****







**REFERENCES**

* Elmasri, R., & Navathe, S. B.  
  *Fundamentals of Database Systems*, 7th Edition, Pearson Education, 2016.
* GeeksforGeeks – DBMS and SQL Articles  
  <https://www.geeksforgeeks.org/dbms/>  
  (Used for understanding normalization, ER modeling, and SQL operations)
* MySQL Documentation  
  <https://dev.mysql.com/doc/>  
  (Official documentation for SQL syntax and MySQL features)
* Silberschatz, A., Korth, H. F., & Sudarshan, S.  
  *Database System Concepts*, 7th Edition, McGraw-Hill Education, 2020.
* W3Schools – SQL Tutorial  
  <https://www.w3schools.com/sql/>  
  (Accessed for learning basic SQL syntax and query examples)