



DIGI-LOCKER APP



A PROJECT REPORT

Submitted by

SAKTHI VISHAL C(2303811724321095)

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 “**DIGI LOCKER APP**” is the bonafide work of **SAKTHI VISHAL C (2303811724321095)** 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 **“DIGI LOCKER APP”** is the result of original work done by me and best of my knowledge, similar work has 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

SAKTHI VISHAL C

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 adequate duration 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 **Mrs.S.GEETHA.M.E.**, Department of **ARTIFICIAL INTELLIGENCE**, for his incalculable suggestions, creativity, assistance and patience which 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 (PSO)

- **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

In an era of digital transformation, the need for a secure, paperless, and efficient method of storing and verifying important documents has become essential. The DigiLocker App is a government-backed digital platform that aims to eliminate the dependency on physical documents by providing citizens with a secure cloud-based space to store and access official documents such as Aadhaar, PAN, driving license, educational certificates, and more. This project proposes the development and analysis of a DigiLocker-like system that enables users to register using their Aadhaar number or mobile number, authenticate through OTP verification, and securely upload or auto-fetch government-issued digital documents. The system ensures document integrity by enabling real-time verification by authorized institutions, thereby reducing the risk of fraud and forgery.

ABSTRACT WITH POs AND PSOs MAPPING

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

ABSTRACT	POs MAPPED	PSOs MAPPED
In an era of digital transformation, the need for a secure, paperless, and efficient method of storing and verifying important documents has become essential. The DigiLocker App is a government-backed digital platform that aims to eliminate the dependency on physical documents by providing citizens with a secure cloud-based space to store and access official documents such as Aadhaar, PAN, driving license, educational certificates, and more.	PO1 -3 PO2 -3 PO4 -3 PO5 -3 PO6 -3 PO7 -3 PO8 -3 PO9 -3 PO10 -3 PO11-3 PO12 -3	PSO1 -3 PSO2 -3

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

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
	1.3 SQL AND DATABASE CONCEPTS	1
2	PROJECT METHODOLOGY	2
	2.1 PROPOSED WORK	2
	2.2 BLOCK DIAGRAM	3
3	MODULE DESCRIPTION	4
	3.1 USER REGISTRATION& MODULE	4
	3.2 DOCUMENT UPLOAD & STORAGE MODULE	4
	3.3 ACCESS & SHARING MODULE	4
	3.4 SECURITY & PRIVACY MODULE	4
4	CONCLUSION & FUTURE SCOPE	5
	APPENDIX A SOURCE CODE	6
	APPENDIX B SCREENSHOTS	12
	REFERENCES	14

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE

The objective of the DigiLocker App project is to design and implement a secure digital platform that enables users to store, access, and share essential documents electronically. The system aims to minimize the reliance on physical documentation by offering a centralized, cloud-based repository integrated with Aadhaar-based authentication and real-time document verification. It focuses on enhancing document security, preventing fraud, simplifying verification processes, and promoting paperless governance in line with national digital initiatives.

1.2 OVERVIEW

The DigiLocker App is a government-supported digital document wallet that offers citizens a secure and convenient way to manage important personal and official documents online. The application allows users to register using their Aadhaar or mobile number, authenticate through OTP, and then upload or auto-fetch government-issued digital documents such as Aadhaar, PAN, driving license, and educational certificates. Documents can be shared securely through QR codes or encrypted links, and verified in real time by partnering institutions. The app is developed with strong encryption, role-based access control, and compliance with data privacy standards to ensure user trust and legal validity:

1.3 SQL AND DATABASE CONCEPTS

A Relational Database Management System (RDBMS) like MySQL, PostgreSQL, or SQLite is suitable due to its structured format and robust query capabilities.

Typical tables in the DigiLocker system might include:

Users: Stores user registration data.

Documents: Stores metadata about uploaded documents.

User_Documents: Maps users to their documents (many-to-many relationship).

Access_Logs: Records access and sharing history for security audits.

CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed DigiLocker App system is aimed at developing a secure, Aadhaar-integrated digital locker platform that enables users to store, manage, and share government-issued and personal documents digitally. This work intends to replicate and enhance the core functionality of the existing DigiLocker ecosystem with added features such as user authentication, document encryption, real-time verification, and seamless access control.

The application will allow users to register using their mobile number or Aadhaar number and log in via OTP-based authentication. Once authenticated, users can upload personal documents or retrieve government-issued digital certificates directly from integrated institutions. These documents will be securely stored in a cloud-based database and can be accessed anytime, anywhere. The system will also include a document sharing module where users can share documents with authorized personnel or institutions through QR codes or secure, encrypted links.

The proposed system will use modern technologies such as cloud storage services, relational databases, secure APIs, and data encryption standards. It will also ensure compliance with digital security and data protection regulations, enabling a reliable, paperless environment that minimizes the risks of document tampering, loss, or forgery.

2.2 BLOCK DIAGRAM



CHAPTER 3

MODULE DESCRIPTION

3.1 USER REGISTRATION & MODULE

This module enables users to create accounts and securely log in to the Digi-Locker app. It manages user credentials with encrypted passwords and handles session management to maintain user login states. Features include email/phone verification during registration, password recovery mechanisms, and optional multi-factor authentication for enhanced security. The module ensures that only authenticated users can access their personal digital lockers and restricts unauthorized access.

3.2 DOCUMENT UPLOAD & STORAGE MODULE

This module allows users to upload their important documents in various formats such as PDF, images, or text files. It handles storing the files securely in the backend, either on the server filesystem or in the database as BLOBs, while maintaining metadata like document type, upload date, and expiry date in the database.

3.3 DOCUMENT VERIFICATION MODULE

This module facilitates the verification of documents uploaded by users. It can incorporate automated checks (such as validating document formats and required fields) as well as manual verification workflows where authorized personnel can approve or reject documents.

3.4 ACCESS & SHARING MODULE

This module enables users to securely share their stored documents with third parties such as government agencies, employers, or educational institutions.

CHAPTER 4

CONCLUSION & FUTURE SCOPE

CONCLUSION

The Digi-Locker App developed using DBMS principles offers a secure, efficient, and user-friendly platform for managing and storing important digital documents. By integrating robust user authentication, seamless document upload and verification, and controlled access and sharing functionalities, the system addresses the need for a reliable digital alternative to traditional physical lockers. The project demonstrates effective use of database management techniques such as relational modeling, secure data storage, and query optimization, ensuring data integrity and quick access. Overall, this solution enhances convenience, reduces paperwork, and improves the security and privacy of personal documents for users.

FUTURE SCOPE

- Integration with Blockchain
- Mobile Application Support
- Advanced Document Analytics.
- Biometric Authentication.
- Cloud Storage Integration.

APPENDIX A SOURCE CODE

```
from flask import Flask, render_template, request, redirect, url_for, flash, session
from werkzeug.utils import secure_filename
import sqlite3
import os
from flask_bcrypt import Bcrypt

app = Flask(__name__)
app.secret_key = 'your-secret-key'
bcrypt = Bcrypt(app)

UPLOAD_FOLDER = 'static/uploads'
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

def init_db():
    conn = sqlite3.connect('users.db')
    c = conn.cursor()
    c.execute("""CREATE TABLE IF NOT EXISTS users
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 username TEXT UNIQUE,
                 password TEXT)""")
    c.execute("""CREATE TABLE IF NOT EXISTS user_details
                (id INTEGER PRIMARY KEY AUTOINCREMENT,
                 user_id INTEGER,
                 aadhar_number TEXT,
                 aadhar_image TEXT,
                 voter_id TEXT,
                 voter_id_image TEXT,
                 education TEXT,
                 FOREIGN KEY(user_id) REFERENCES users(id))""")
```



```
conn.commit()
```

```
conn.close()
```

```
def allowed_file(filename):
```

```
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```
@app.route('/')
```

```
def index():
```

```
    if 'user_id' in session:
```

```
        return redirect(url_for('dashboard'))
```

```
    return render_template('login.html')
```

```
@app.route('/register', methods=['GET', 'POST'])
```

```
def register():
```

```
    if request.method == 'POST':
```

```
        username = request.form['username']
```

```
        password = bcrypt.generate_password_hash(request.form['password']).decode('utf-8')
```

```
    try:
```

```
        conn = sqlite3.connect('users.db')
```

```
        c = conn.cursor()
```

```
        c.execute("INSERT INTO users (username, password) VALUES (?, ?)",
```

```
                  (username, password))
```

```
        conn.commit()
```

```
        flash('Registration successful! Please login.', 'success')
```

```
        return redirect(url_for('index'))
```

```
    except sqlite3.IntegrityError:
```

```
        flash('Username already exists!', 'error')
```

```
    finally:
```

```
        conn.close()
```

```

return render_template('register.html')

@app.route('/login', methods=['POST'])
def login():
    username = request.form['username']
    password = request.form['password']

    conn = sqlite3.connect('users.db')
    c = conn.cursor()
    c.execute("SELECT * FROM users WHERE username = ?", (username,))
    user = c.fetchone()
    conn.close()

    if user and bcrypt.check_password_hash(user[2], password):
        session['user_id'] = user[0]
        flash('Login successful!', 'success')
        return redirect(url_for('dashboard'))

    flash('Invalid credentials!', 'error')
    return redirect(url_for('index'))

@app.route('/dashboard')
def dashboard():
    if 'user_id' not in session:
        return redirect(url_for('index'))

    conn = sqlite3.connect('users.db')
    c = conn.cursor()
    c.execute("SELECT * FROM user_details WHERE user_id = ?", (session['user_id'],))
    details = c.fetchone()
    conn.close()

```

```
return render_template('dashboard.html', details=details)
```

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

```
def save_details():
```

```
    if 'user_id' not in session:
```

```
        return redirect(url_for('index'))
```

```
    aadhar_number = request.form['aadhar_number']
```

```
    voter_id = request.form['voter_id']
```

```
    education = request.form['education']
```

```
    aadhar_image = None
```

```
    voter_id_image = None
```

```
    if 'aadhar_image' in request.files:
```

```
        file = request.files['aadhar_image']
```

```
        if file and allowed_file(file.filename):
```

```
            filename = secure_filename(file.filename)
```

```
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
```

```
            aadhar_image = filename
```

```
    if 'voter_id_image' in request.files:
```

```
        file = request.files['voter_id_image']
```

```
        if file and allowed_file(file.filename):
```

```
            filename = secure_filename(file.filename)
```

```
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
```

```
            voter_id_image = filename
```

```
    conn = sqlite3.connect('users.db')
```

```
    c = conn.cursor()
```

```

c.execute("SELECT * FROM user_details WHERE user_id = ?", (session['user_id'],))
existing_details = c.fetchone()

if existing_details:
    c.execute("""UPDATE user_details
                SET aadhar_number = ?, aadhar_image = ?, voter_id = ?, voter_id_image = ?,
education = ?
                WHERE user_id = ?""",
                (aadhar_number, aadhar_image, voter_id, voter_id_image, education,
session['user_id']))
else:
    c.execute("""INSERT INTO user_details
                (user_id, aadhar_number, aadhar_image, voter_id, voter_id_image, education)
                VALUES (?, ?, ?, ?, ?, ?)""",
                (session['user_id'], aadhar_number, aadhar_image, voter_id, voter_id_image,
education))

conn.commit()
conn.close()

flash('Details saved successfully!', 'success')
return redirect(url_for('dashboard'))

@app.route('/logout')
def logout():
    session.pop('user_id', None)
    flash('Logged out successfully!', 'success')
    return redirect(url_for('index'))

```

```
if __name__ == '__main__':  
    os.makedirs(UPLOAD_FOLDER, exist_ok=True)  
    init_db()  
app.run(debug=True)safef);
```

APPENDIX B SCREENSHOTS

DIGI-LOCKER

Registration successful! Please login.

Login

Username

Password

Login

Don't have an account? [Register](#)

User Dashboard

[Logout](#)

Aadhar Number

Aadhar Image

No file chosen

Voter ID

Voter ID Image

No file chosen

Education Details

Save Details

REFERENCES

- 1.DigiLocker Official Website - <https://digilocker.gov.in>
2. Government of India's Digital India Initiative (DigiLocker) -
<https://www.digitalindia.gov.in/digilocker>
- 3.DigiLocker API Documentation - <https://api.digitallocker.gov.in/>