Project Documentation: Screen Recorder with Flask & MySQL

1. Introduction

This project is a **web-based screen recorder** built using **Flask, HTML, CSS (Tailwind), and MySQL**. It allows users to:
✅ Record their screen.
✅ Preview and download recordings.
✅ Store videos in a MySQL database.
✅ Secure authentication with login and signup.

It is a simple yet efficient solution for **online tutorials, presentations, and user support videos**.

---

2. Features

🔹 User Authentication

- Signup & login functionality with a **secure user database**.
- Credentials stored in MySQL for security.

🔹 Screen Recording

- Capture **entire screen, app window, or tab**.
- High-quality **WebM format** video output.

🔹 Video Management

- **Auto-downloads after recording**.
- Saves videos in a **dedicated folder (Videos/)**.
- Video list stored in MySQL for easy retrieval.

🔹 UI & UX Enhancements

- **Tailwind CSS** for a modern look.
- **Animations** for smooth transitions.
- **Responsive design** for all devices.

---

3. Technology Stack

Technology	Purpose
Python (Flask)	Backend web framework
HTML, CSS (Tailwind)	Frontend UI/UX
JavaScript	Handles recording functionality
MySQL	Database for storing users & videos
Flask-MySQLdb	Connector for MySQL

---

4. Database Schema

The project uses **two tables: user and videos**.

🔹 User Table (user)

Stores user credentials.

```
CREATE TABLE user (
    username VARCHAR(100) PRIMARY KEY,
    password VARCHAR(255) NOT NULL
);
```

🔹 Videos Table (videos)

Stores recorded video file paths.

```
CREATE TABLE videos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    filename VARCHAR(255) NOT NULL,
    upload_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

---

5. System Flow

1 **User Registration & Login**

- User **signs up** (credentials saved in MySQL).
- User **logs in** and accesses the dashboard.

2 **Screen Recording Process**

- User clicks "Start Recording" → Captures screen.
- Clicks "Stop Recording" → Video is saved.
- **Alert appears** → User presses Enter → Video auto-downloads.
- Video stored in **"Videos/" folder**.

3 **Video Management**

- Videos **listed in recordings page**.
- Users can **download** or **delete** old recordings.

---

6. Implementation Details

🔹 Backend (Flask)

Handles **user authentication**, **video storage**, and **database operations**.

server.py (Main Flask Application)

- **Routes:**
  - /signup → Registers new users.
  - /login → User authentication.
  - /dashboard → User dashboard.
  - /recorder → Screen recording page.
  - /upload → Saves recorded video.
  - /recordings → Displays saved videos.
  - /delete/<filename> → Deletes video file.

🔹 Frontend (HTML, CSS, JavaScript)

- **Tailwind CSS** → Enhances UI with animations & responsive design.
- **JavaScript** → Handles screen recording and auto-download.

---

7. How to Run the Project

🔹 Prerequisites

✅ Install **Python 3**
✅ Install **MySQL**
✅ Install dependencies

```
pip install flask flask-mysqldb
```

🔹 Set Up Database

```
CREATE DATABASE YA;
USE YA;

CREATE TABLE user (
    username VARCHAR(100) PRIMARY KEY,
    password VARCHAR(255) NOT NULL
);

CREATE TABLE videos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    filename VARCHAR(255) NOT NULL,
    upload_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

🔹 Run Flask Server

```
python server.py
```

Visit: **http://127.0.0.1:5000/**

---

8. Future Enhancements

✅ **Password Hashing:** Encrypt user passwords for security.
✅ **User-Based Videos:** Store videos per user instead of globally.
✅ **Cloud Storage:** Upload videos to cloud instead of local storage.
✅ **Live Streaming:** Add real-time screen sharing.

📌 Summary

This project provides a **full-fledged screen recorder** with a **modern UI, database storage, and auto-download feature**. It can be used for **tutorials, online classes, or support videos**. 🚀🔥