Video Recorder Project – Documentation

1. Project Overview

The **Video Recorder Project** is a web-based application that allows users to record their screen, preview recordings, and download them. It includes a login/signup system to manage user access. The project is built using **Python (Flask) for the backend** and **HTML, Tailwind CSS, and JavaScript for the frontend**.

---

2. Technologies Used

- **Backend:** Python (Flask)
- **Frontend:** HTML, Tailwind CSS, JavaScript
- **Database:** `users.txt` (for storing login credentials)
- **Screen Recording:** MediaRecorder API

---

3. Project Structure

```
/video_recorder_project
│── /templates
│   ├── index.html
│   ├── website.html
│   ├── login.html
│   ├── signup.html
│   ├── dashboard.html
│   ├── recorder.html
│   ├── recordings.html
│── /static
│   ├── styles.css (if needed)
│── users.txt
│── server.py
│── README.md
```

---

4. Features

Frontend Features

✅ Welcome Page (`index.html`)

- Animated text and background effect
- Auto-redirects to `website.html` after 3 seconds

✅ Main Website (`website.html`)

- Hero section with welcome text
- Features list (Screen Recording, Download, Preview)
- Login & Signup buttons

✅ Login Page (`login.html`)

- Takes **username & password** input
- Sends login data to `server.py`
- Redirects to `dashboard.html` if login is successful

✅ Signup Page (`signup.html`)

- Allows new user registration
- Stores credentials in `users.txt`

✅ Dashboard (`dashboard.html`)

- Displays **Welcome Message**
- "Start Recording" button (Links to `recorder.html`)

✅ Screen Recorder (`recorder.html`)

- Uses **MediaRecorder API** to record the screen
- Allows preview & download of the recording

✅ Recordings Page (`recordings.html`)

- Displays previously recorded videos stored in **localStorage**
- Users can **delete recordings**

---

5. Backend (server.py)

Handles:

- ✅ User login & authentication (checks `users.txt`)

- ✓ User signup (stores new users in `users.txt`)
- ✓ Serves all HTML pages

Code Snippet (server.py)

```
# Route for Login Page
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        with open("users.txt", "r") as file:
            users = file.readlines()
        for user in users:
            stored_username, stored_password = user.strip().split(',')
            if username == stored_username and password ==
stored_password:
                return redirect('/dashboard')
        return "Invalid Credentials"
    return render_template('login.html')
```

---

6. How to Run the Project

Step 1: Install Flask

If Flask is not installed, install it using:

```
pip install flask
```

Step 2: Run the Flask Server

Navigate to the project folder and run:

```
python server.py
```

Step 3: Open in Browser

Go to: **`http://127.0.0.1:5000/`**

---

7. User Credentials (Stored in users.txt)

The following users exist in `users.txt`:

admin,admin123
yukesh,ya
user1,password1

New users can also register via the Signup page.

---

8. UI/UX Design

The project uses **Tailwind CSS** for a **modern look**.

- White background with blue buttons (same as `website.html`).
- Styled **login & signup pages** to match the design.
- **Buttons & forms** are neatly aligned.

---

9. Troubleshooting

Issue	Solution
404 Not Found	Make sure the route exists in `server.py`
405 Method Not Allowed	Ensure the correct HTTP method (`POST` or `GET`) is used
Login not working	Check `users.txt` for correct credentials
Signup not saving users	Ensure `users.txt` exists and has write permissions

---

10. Future Improvements

- ✅ Store recordings in a database instead of localStorage.
- ✅ Add a **profile page** for users.
- ✅ Improve **recording controls (pause, resume, trim)**.
- ✅ Allow **recording playback** inside the website.

---

11. Conclusion

The **Video Recorder Project** provides an easy way to **record the screen, save recordings, and manage user access**. It combines **Flask for the backend**, **HTML + Tailwind CSS for the UI**, and the **MediaRecorder API** for screen recording.