

Public School Website (Flask) — Full Project

This is a complete Flask project with the pages/features you asked for: **Home, Photo/Video Upload (with gallery), Results (PDF uploads), Motivation, Contact (phone & email), Admissions (form saved to CSV).**

Project Structure

```
school_site/
├─ app.py
├─ requirements.txt
├─ Procfile
├─ render.yaml           # optional (for Render.com)
├─ .env.example          # optional, local config sample
├─ .gitignore
├─ static/
│  ├─ uploads/           # auto-created; holds photos/videos/results
│  ├─ css/
│  │  └─ styles.css
│  └─ img/
│     └─ logo.png        # put your school logo here
├─ templates/
│  ├─ base.html
│  ├─ index.html         # Home
│  ├─ motivation.html    # Motivation/About
│  ├─ results.html       # List + upload results PDFs
│  ├─ upload.html        # Photo/Video upload + gallery
│  ├─ contact.html       # Contact
│  └─ admissions.html    # Admission form
└─ data/
   └─ admissions.csv     # auto-created on first submission
```

app.py

```
import os
from datetime import datetime
from flask import Flask, render_template, request, redirect, url_for, flash,
send_from_directory
from werkzeug.utils import secure_filename

BASE_DIR = os.path.abspath(os.path.dirname(__file__))
```

```

UPLOAD_DIR = os.path.join(BASE_DIR, 'static', 'uploads')
DATA_DIR = os.path.join(BASE_DIR, 'data')

ALLOWED_IMAGE_EXT = {"png", "jpg", "jpeg", "gif"}
ALLOWED_VIDEO_EXT = {"mp4", "mov", "avi", "mkv", "webm"}
ALLOWED_RESULT_EXT = {"pdf"}
MAX_CONTENT_MB = 100

app = Flask(__name__)
app.config['SECRET_KEY'] = os.getenv('SECRET_KEY', 'dev-key-change-me')
app.config['UPLOAD_FOLDER'] = UPLOAD_DIR
app.config['MAX_CONTENT_LENGTH'] = MAX_CONTENT_MB * 1024 * 1024

os.makedirs(UPLOAD_DIR, exist_ok=True)
os.makedirs(DATA_DIR, exist_ok=True)

# ----- helpers -----

def allowed_file(filename, allowset):
    return "." in filename and filename.rsplit('.', 1)[1].lower() in allowset

def save_upload(file_storage, subfolder="media"):
    folder = os.path.join(UPLOAD_DIR, subfolder)
    os.makedirs(folder, exist_ok=True)
    filename = secure_filename(file_storage.filename)
    timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
    # avoid overwriting: timestamp prefix
    final_name = f"{timestamp}_{filename}"
    path = os.path.join(folder, final_name)
    file_storage.save(path)
    return final_name, folder

# ----- routes -----

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/motivation')
def motivation():
    return render_template('motivation.html')

@app.route('/contact')
def contact():

```

```

        return render_template('contact.html')

@app.route('/admissions', methods=['GET', 'POST'])
def admissions():
    csv_path = os.path.join(DATA_DIR, 'admissions.csv')
    if request.method == 'POST':
        name = request.form.get('name', '').strip()
        father = request.form.get('father', '').strip()
        grade = request.form.get('grade', '').strip()
        phone = request.form.get('phone', '').strip()
        email = request.form.get('email', '').strip()
        message = request.form.get('message', '').strip()

        if not name or not grade or not phone:
            flash('Please fill at least Name, Grade, and Phone.', 'error')
            return redirect(url_for('admissions'))

        is_new = not os.path.exists(csv_path)
        with open(csv_path, 'a', encoding='utf-8') as f:
            if is_new:
                f.write('timestamp,name,father,grade,phone,email,message\n')
                f.write(f"{datetime.now().isoformat(timespec='seconds')},{name},{father},{grade},{phone},{email},{message}\n")
            flash('Application submitted successfully. We will contact you soon.', 'success')
            return redirect(url_for('admissions'))

    return render_template('admissions.html')

@app.route('/results', methods=['GET', 'POST'])
def results():
    results_folder = os.path.join(UPLOAD_DIR, 'results')
    os.makedirs(results_folder, exist_ok=True)

    if request.method == 'POST':
        file = request.files.get('result_pdf')
        if not file or file.filename == '':
            flash('No file selected.', 'error')
            return redirect(url_for('results'))
        if not allowed_file(file.filename, ALLOWED_RESULT_EXT):
            flash('Only PDF files are allowed.', 'error')
            return redirect(url_for('results'))
        saved_name, _ = save_upload(file, subfolder='results')
        flash(f'Result uploaded: {saved_name}', 'success')
        return redirect(url_for('results'))

```

```

# list PDFs
pdfs = []
for fname in sorted(os.listdir(results_folder)):
    if allowed_file(fname, ALLOWED_RESULT_EXT):
        pdfs.append(fname)
return render_template('results.html', pdfs=pdfs)

@app.route('/upload', methods=['GET', 'POST'])
def upload():
    media_folder = os.path.join(UPLOAD_DIR, 'media')
    os.makedirs(media_folder, exist_ok=True)

    if request.method == 'POST':
        file = request.files.get('file')
        if not file or file.filename == '':
            flash('No file selected.', 'error')
            return redirect(url_for('upload'))
        ext = (file.filename.rsplit('.', 1)[1].lower() if '.' in file.filename
else '')
        if ext in ALLOWED_IMAGE_EXT.union(ALLOWED_VIDEO_EXT):
            saved_name, _ = save_upload(file, subfolder='media')
            flash(f'Uploaded: {saved_name}', 'success')
        else:
            flash('File type not allowed.', 'error')
            return redirect(url_for('upload'))

# list gallery
gallery = []
for fname in sorted(os.listdir(media_folder)):
    path = os.path.join('uploads', 'media', fname)
    ext = fname.rsplit('.', 1)[1].lower() if '.' in fname else ''
    kind = 'video' if ext in ALLOWED_VIDEO_EXT else 'image'
    gallery.append({'name': fname, 'path': path, 'type': kind})
return render_template('upload.html', gallery=gallery)

# optional direct file serving for results
@app.route('/uploads/results/<path:filename>')
def downloads(filename):
    return send_from_directory(os.path.join(app.config['UPLOAD_FOLDER'],
'results'), filename, as_attachment=False)

# error handling for large files
@app.errorhandler(413)
def too_large(e):
    flash(f'File too large. Max {MAX_CONTENT_MB} MB.', 'error')

```

```

        return redirect(request.url)

if __name__ == '__main__':
    app.run(debug=True)

```



templates/base.html

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>{% block title %}My School{% endblock %}</title>
    <!-- Tailwind via CDN for quick styling -->
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/
styles.css') }}">
</head>
<body class="bg-gray-50 text-gray-800">
    <header class="bg-white shadow sticky top-0 z-50">
        <div class="max-w-6xl mx-auto px-4 py-3 flex items-center justify-between">
            <a href="{{ url_for('home') }}" class="flex items-center gap-3">
                
                <span class="font-bold text-xl">My School</span>
            </a>
            <nav class="flex gap-5 text-sm">
                <a href="{{ url_for('home') }}" class="hover:text-blue-600">Home</a>
                <a href="{{ url_for('motivation') }}" class="hover:text-
blue-600">Motivation</a>
                <a href="{{ url_for('results') }}" class="hover:text-blue-600">Results</
a>
                <a href="{{ url_for('upload') }}" class="hover:text-blue-600">Upload</a>
                <a href="{{ url_for('admissions') }}" class="hover:text-
blue-600">Admissions</a>
                <a href="{{ url_for('contact') }}" class="hover:text-blue-600">Contact</
a>
            </nav>
        </div>
    </header>

    <main class="max-w-6xl mx-auto px-4 py-8">
        {% with messages = get_flashed_messages(with_categories=true) %}

```

```

        {% if messages %}
        <div class="space-y-2 mb-6">
            {% for cat, msg in messages %}
            <div class="p-3 rounded border {{ 'border-green-300 bg-green-50 text-
green-700' if cat=='success' else 'border-red-300 bg-red-50 text-red-700' }}">
                {{ msg }}
            </div>
            {% endfor %}
        </div>
        {% endif %}
    {% endwith %}

    {% block content %}{% endblock %}
</main>

<footer class="border-t bg-white">
    <div
class="max-w-6xl mx-auto px-4 py-6 text-sm text-gray-600 flex flex-wrap items-
center justify-between gap-3">
        <p>© {{ 2025 }} My School. All rights reserved.</p>
        <p>Made with Flask.</p>
    </div>
</footer>
</body>
</html>

```



templates/index.html

```

{% extends 'base.html' %}
{% block title %}Home - My School{% endblock %}
{% block content %}
<section class="grid md:grid-cols-2 gap-8 items-center">
    <div>
        <h1 class="text-3xl md:text-4xl font-bold mb-4">Welcome to My School</h1>
        <p class="mb-4">A place of learning, growth, and community. Explore
admissions, check results, and see our latest moments in the gallery.</p>
        <div class="flex gap-3">
            <a href="{{ url_for('admissions') }}"
class="px-4 py-2 rounded bg-blue-600 text-white">Apply Now</a>
            <a href="{{ url_for('results') }}" class="px-4 py-2 rounded border">View
Results</a>
        </div>
    </div>
    <div class="aspect-video rounded-xl overflow-hidden shadow">

```

```

        <video controls class="w-full h-full object-cover">
            <source src="{% url_for('static', filename='uploads/media/sample.mp4') %}"
type="video/mp4">
        </video>
    </div>
</section>

<section class="mt-12">
    <h2 class="text-2xl font-semibold mb-4">Latest Highlights</h2>
    <div class="grid grid-cols-2 md:grid-cols-4 gap-4">
        {% for item in ['img1.jpg','img2.jpg','img3.jpg','img4.jpg'] %}
            <div class="aspect-square rounded-lg bg-gray-200 flex items-center
justify-center">Add photos</div>
        {% endfor %}
    </div>
</section>
{% endblock %}

```



templates/motivation.html

```

{% extends 'base.html' %}
{% block title %}Motivation - My School{% endblock %}
{% block content %}
<h1 class="text-3xl font-bold mb-4">Our Mission & Motivation</h1>
<p class="mb-4">Empowering students with knowledge, character, and skills for
the future.</p>
<blockquote class="p-4 border-1-4 bg-blue-50 border-blue-400 rounded">
    "Education is the most powerful weapon which you can use to change the world."
    - Nelson Mandela
</blockquote>

<h2 class="text-xl font-semibold mt-8 mb-2">Staff Gallery</h2>
<p class="text-sm text-gray-600 mb-4">Add staff photos in <code>static/uploads/
media</code> to display them on the Upload/Gallery page.</p>
{% endblock %}

```



templates/results.html

```

{% extends 'base.html' %}
{% block title %}Results - My School{% endblock %}
{% block content %}

```

```

<div class="flex items-center justify-between mb-4">
  <h1 class="text-2xl font-bold">Results</h1>
</div>

<form action="{{ url_for('results') }}" method="post" enctype="multipart/form-
data" class="p-4 border rounded mb-6">
  <label class="block mb-2 font-medium">Upload Result (PDF only)</label>
  <input type="file" name="result_pdf" accept="application/pdf" class="mb-3" />
  <button class="px-4 py-2 rounded bg-blue-600 text-white">Upload</button>
  <p class="text-xs text-gray-500 mt-2">Max size: 100MB</p>
</form>

<ul class="space-y-2">
  {% if pdfs %}
    {% for pdf in pdfs %}
      <li class="p-3 border rounded flex items-center justify-between">
        <span>{{ pdf }}</span>
        <a href="{{ url_for('downloads', filename=pdf) }}" target="_blank"
class="text-blue-600">Open</a>
      </li>
    {% endfor %}
  {% else %}
    <li class="text-gray-600">No results uploaded yet.</li>
  {% endif %}
</ul>
{% endblock %}

```



templates/upload.html

```

{% extends 'base.html' %}
{% block title %}Upload & Gallery - My School{% endblock %}
{% block content %}
<h1 class="text-2xl font-bold mb-4">Upload Photo/Video</h1>
<form action="{{ url_for('upload') }}" method="post" enctype="multipart/form-
data" class="p-4 border rounded mb-6">
  <input type="file" name="file" class="mb-3" />
  <button class="px-4 py-2 rounded bg-blue-600 text-white">Upload</button>
  <p class="text-xs text-gray-500 mt-2">Allowed: images (png, jpg, jpeg, gif) &
videos (mp4, mov, avi, mkv, webm). Max 100MB.</p>
</form>

<h2 class="text-xl font-semibold mb-4">Gallery</h2>
<div class="grid grid-cols-2 md:grid-cols-4 gap-4">
  {% for item in gallery %}

```



```

<div class="rounded overflow-hidden border">
  {% if item.type == 'video' %}
    <video controls class="w-full aspect-square object-cover">
      <source src="{{ url_for('static', filename=item.path) }}">
    </video>
  {% else %}
    
  {% endif %}
  <div class="p-2 text-xs text-gray-600 truncate">{{ item.name }}</div>
</div>
{% endfor %}
</div>
{% endblock %}

```



templates/contact.html

```

{% extends 'base.html' %}
{% block title %}Contact - My School{% endblock %}
{% block content %}
<h1 class="text-2xl font-bold mb-4">Contact Us</h1>
<div class="grid md:grid-cols-2 gap-6">
  <div class="space-y-2">
    <p><strong>Phone:</strong> +92-300-0000000</p>
    <p><strong>Email:</strong> school@example.com</p>
    <p><strong>Address:</strong> Your City, Pakistan</p>
    <div class="rounded overflow-hidden">
      <!-- Optional: embed Google Map iframe here -->
      <div class="p-6 border rounded bg-gray-50 text-sm">Map placeholder</div>
    </div>
  </div>
  <form class="p-4 border rounded">
    <p class="text-sm text-gray-500 mb-2">(Optional demo form – add backend
later if needed)</p>
    <input class="w-full border rounded p-2 mb-3" placeholder="Your Name" />
    <input class="w-full border rounded p-2 mb-3" placeholder="Email" />
    <textarea class="w-full border rounded p-2 mb-3" placeholder="Message"
rows="4"></textarea>
    <button class="px-4 py-2 rounded bg-blue-600 text-white"
type="button">Send</button>
  </form>
</div>
{% endblock %}

```



templates/admissions.html

```
{% extends 'base.html' %}
{% block title %}Admissions - My School{% endblock %}
{% block content %}
<h1 class="text-2xl font-bold mb-4">Admission Form</h1>
<form action="{{ url_for('admissions') }}" method="post" class="grid md:grid-
cols-2 gap-4 p-4 border rounded">
  <input class="border rounded p-2" name="name" placeholder="Student Name *"
required>
  <input class="border rounded p-2" name="father" placeholder="Father/Guardian
Name">
  <select class="border rounded p-2" name="grade" required>
    <option value="">Select Grade *</option>
    <option>Nursery</option>
    <option>KG</option>
    <option>Class 1</option>
    <option>Class 2</option>
    <option>Class 3</option>
    <option>Class 4</option>
    <option>Class 5</option>
    <option>Class 6</option>
    <option>Class 7</option>
    <option>Class 8</option>
    <option>Class 9</option>
    <option>Class 10</option>
  </select>
  <input class="border rounded p-2" name="phone" placeholder="Phone *" required>
  <input class="border rounded p-2" type="email" name="email"
placeholder="Email">
  <textarea class="md:col-span-2 border rounded p-2" name="message" rows="4"
placeholder="Message"></textarea>
  <div class="md:col-span-2">
    <button class="px-4 py-2 rounded bg-blue-600 text-white">Submit
Application</button>
  </div>
</form>
<p class="text-sm text-gray-500 mt-3">Submissions are saved to <code>data/
admissions.csv</code>.</p>
{% endblock %}
```



static/css/styles.css

```
/* Add any custom styles here */
```



requirements.txt

```
Flask==3.0.3  
Werkzeug==3.0.3  
python-dotenv==1.0.1
```



Procfile (for Render/Railway/Heroku-style)

```
web: gunicorn app:app
```

If you use the Procfile above, add `gunicorn` to requirements:

```
Gunicorn==22.0.0
```



.gitignore

```
__pycache__/  
*.pyc  
.env  
.env.*  
instance/  
static/uploads/  
data/
```



render.yaml (optional, for Render.com)

```
services:  
- type: web  
  name: school-site
```

```
env: python
plan: free
buildCommand: pip install -r requirements.txt
startCommand: gunicorn app:app
autoDeploy: true
```



`.env.example` (optional)

```
SECRET_KEY=change-this-secret
```



How to Run Locally

```
# 1) create folder and files as above
python -m venv venv
source venv/bin/activate # Windows: venv\Scripts\activate
pip install -r requirements.txt
python app.py
# open: http://127.0.0.1:5000
```



Deploy Free on Render (public link)

1. Push this project to **GitHub**.
2. Create a new **Web Service** on **Render.com** → connect your repo.
3. Runtime: Python; Build Command: `pip install -r requirements.txt`
4. Start Command: `gunicorn app:app`
5. Add environment variable **SECRET_KEY** (any random string).
6. Deploy → get your public URL.



Notes

- Uploaded files go into `static/uploads/` and are immediately visible in the gallery/results.
- Admissions are logged to `data/admissions.csv` (kept out of Git by `.gitignore`).
- You can rename **My School** and replace the **logo.png**.
- To limit who can upload (admin only), you can add a simple password on the upload/results forms later; I can add that for you if needed.