

The background of the entire page is a white canvas with large, flowing, overlapping shapes in various shades of teal and green. These shapes create a sense of movement and depth, framing the central text.

FUNNEL PROJECT

How to do it yourself and save money

Miika Ruutinen

Project name: Funnel Project – Simple Email Funnel with Flask

Developer: Miika Ruutinen

Technologies used: Flask, SQLite, Python, HTML/CSS/JS, Gmail SMTP, Render (hosting), GitHub

Goal:

Build a web page where users can subscribe to an email list. Once subscribed, they receive a welcome email automatically.

Key Features:

- ✓ **Responsive landing page (HTML/CSS)**
 - ✓ **AJAX subscription form**
- ✓ **Email addresses saved to SQLite database**
- ✓ **Automatic welcome email via Gmail SMTP**
 - ✓ **Public deployment via Render**

Folder Layout:

```
/funnel-projekti/
├─ app.py                # Flask backend logic
├─ templates/
│   └─ index.html        # Main HTML template
├─ static/
│   ├─ style.css          # CSS styles
│   └─ script.js          # JS for handling the form
├─ media/
│   └─ pexels-goumbik.jpg # Image on the landing page
├─ .env                  # Environment variables (EMAIL_USER, EMAIL_PASS)
├─ requirements.txt       # Python dependencies
└─ subscriptions.db       # SQLite database (created on runtime)
```

Core Libraries:

Flask – Web framework
smtplib & email.mime – Email sending
sqlite3 – Local database
dotenv – Loading environment variables
gunicorn – WSGI server for deployment on Render

Page 3: Code Examples

Flask route for handling subscriptions:

```
python

@app.route("/subscribe", methods=["POST"])
def subscribe():
    data = request.get_json()
    email = data.get("email")
    if not email:
        return "Missing email", 400

    try:
        conn = sqlite3.connect("subscriptions.db")
        c = conn.cursor()
        c.execute("INSERT INTO subscribers (email) VALUES (?)", (email,))
        conn.commit()
    except sqlite3.IntegrityError:
        return "Email already registered.", 400
    finally:
        conn.close()

    send_welcome_email(email)
    return "Thanks for subscribing!", 200
```

Sending email via Gmail SMTP:

```
python

def send_welcome_email(to_email):
    subject = "Welcome to our newsletter!"
    body = "Thank you for subscribing. Exciting things ahead!"

    msg = MIMEText(body)
    msg["Subject"] = subject
    msg["From"] = EMAIL
    msg["To"] = to_email

    with smtplib.SMTP_SSL("smtp.gmail.com", 465) as server:
        server.login(EMAIL, PASSWORD)
        server.send_message(msg)
```

Page 4: Deployment & Usage

1. Local Testing:

```
$ python app.py
```

Then open the app at: <http://127.0.0.1:5000>

2. Push to GitHub:

```
$ git add .
```

```
$ git commit -m "Initial commit"
```

```
$ git push origin main
```

3. Deploy on Render:

Create a new Web Service on render.com.

Connect your GitHub repo.

Set Build Command: `pip install -r requirements.txt`

Set Start Command: `gunicorn app:app`

Add environment variables:

`EMAIL_USER` = your Gmail address

`EMAIL_PASS` = your Gmail App Password

4. Done!

✓ Website is now live at: <https://funnel-projekti.onrender.com>

✓ Users can subscribe and receive automatic welcome emails.