

# Project Documentation: Automated Google Form Submission and Email Notification

## Overview

This project is a Django-based web application designed to automate the process of filling out a Google Form using Selenium WebDriver and send a screenshot of the submission via email.

## Project Structure

- **views.py**: Contains the view function to handle the form submission process.
- **urls.py**: URL routing for the form view.
- **settings.py**: Configuration for email settings.

## Requirements

- Django
- Selenium
- Google Chrome browser
- Email backend configured in Django settings

## Installation

1. Install dependencies:

```
pip install -r requirements.txt
```

*requirements.txt* has all the libraires required for this project

2. Configure email backend in *settings.py*:

```
EMAIL_BACKEND = "django.core.mail.backends.smtp.EmailBackend"
EMAIL_HOST = "smtp.gmail.com"
EMAIL_PORT = 587
EMAIL_USE_TLS = True
DEFAULT_FROM_EMAIL = "MY APP"
EMAIL_HOST_USER = os.environ.get("AA_EMAIL_HOST_USER")
EMAIL_HOST_PASSWORD = os.environ.get("AA_EMAIL_HOST_PASSWORD")
```

## Usage

1. Activate environment:  
Open new terminal > type `.\venv\Scripts\activate` and enter
2. Run Django server:  
*python manage.py runserver*

3. Access the form submission endpoint:  
Open your web browser and navigate to <http://127.0.0.1:8000>

## Detailed Functionality

*form(request)*

This function automates the submission of a Google Form and sends a screenshot of the submission via email.

1. Initialize Selenium WebDriver:

```
options = Options()
driver = webdriver.Chrome(options=options)
```

2. Navigate to the Google Form URL:

```
f_url = 'https://forms.gle/WT68aV5UnPajeoSc8'
driver.get(f_url)
```

3. Fill out the form fields:

```
driver.find_element("xpath", '//input_xpath').send_keys("value")
```

4. Submit the form:

```
driver.find_element("xpath", '//submit_button_xpath').click()
```

5. Take a screenshot:

```
driver.save_screenshot('ss.png')
```

6. Send an email with the screenshot:

```
email = EmailMessage( subject='Subject', body='Body',
from_email='from@example.com', to=['to@example.com'], cc=
['cc@example.com'], ) email.attach_file("ss.png") email.send(fail_silently=False)
```

7. Close the webdriver:

```
driver.quit()
```

## Error Handling

If any exception occurs during the form submission process, it will be caught, and an error message will be returned in the HTTP response.