

# Thilakshana Weerasinghe

## Python Developer

---

Phone: +94 712835711  
Email: [tthiyural@gmail.com](mailto:tthiyural@gmail.com)  
Address: 15/100, Seevali Lane, Borella.  
GitHub : [Click-me](#)  
Web : [Click-me](#)  
Designing: [Click-me](#)

---

### SKILLS

- JavaScript, HTML5, CSS3
  - Django, Flask
  - MySQL, MongoDB, SQLite, PostgreSQL, MongoDB
  - Git, **GitHub** and Version Control
  - RESTful API Development (Django)
  - Python Automation and Web Scraping  
Selenium, beautiful soup
  - [Graphic design](#)
- 

### EDUCATION

Bachelor of Science(IM), *Undergraduate* Sri Lanka Institute of Information Technology (SLIIT), Malabe, Sri Lanka.

---

### CERTIFICATIONS/LICENSES

- Completed [100 Days of Code – Python Bootcamp \(Udemy\)](#) one of the most comprehensive Python courses on the web, covering real- world projects, automation, APIs, and OOP.
- AI/ML stage 1 -SLIIT
- completed 15 Digital Marketing certifications ([Link](#)) *Google Skill Shop*
- Marketing and digital strategy Certifications ([Link](#)) *HubSpot*

### SUMMARY

**Aspiring Full Stack Developer** actively looking for opportunities to apply and grow my skills in real-world projects. I've completed 50+ Python projects through the *100 Days of Code* challenge and built 8+ full-stack web apps using Django, Flask, and PostgreSQL.

I'm passionate about creating smart, scalable web applications and have a strong interest in integrating AI/ML into web solutions. I enjoy writing clean, maintainable code and thrive in collaborative, innovative environments.

### PROJECTS

#### Study Buddy – Full stack project

**Tech Stack:** HTML5, CSS3, TensorFlow, Dialog flow, Django

**Link:** [Study Buddy Repository](#)

- Implemented Django **templating and view logic** to deliver dynamic content and provide a user-friendly interface for academic support.
- Managed routing, form handling, and data flow between the chatbot and frontend, ensuring a smooth and scalable user experience.
- backend optimization and ensured clean, maintainable code aligned with Django best practices.

#### E-Commerce Web Application

**Tech Stack:** Python, Django, Stripe API

**Link:** [E-Commerce Rest API-Django Repository](#)

- Developed the **Django backend** with a RESTful architecture to handle products, user authentication, order management, and cart functionalities.
- Implemented **secure payment processing** using the Stripe API, including order creation, transaction handling, and real-time payment status updates.
- Created clean, modular views and serializers to ensure scalable and maintainable code structure.

#### Image Processing Project – Academic Y3S1

**Tech Stack:** Python, OpenCV2, NumPy, Matplotlib

**Links:** [Image Processing Repository](#)

- Contributed as a **core developer** in a team project focused on building a modular image processing tool using Python.
- Personally implemented key features including **grayscale conversion, blurring, thresholding, and contour detection** using OpenCV and NumPy.
- Designed and tested **custom filters and edge detection modules**, helping the team experiment with different image enhancement techniques.

#### Personal Web Project - [Visit](#)

**Tech Stack:** HTML5, CSS3, JavaScript, Flask

**Links:** [Personal Web Repository](#)

- Designed and developed a **personal portfolio website** to professionally showcase projects, designs, and development work.
- Created a **visually appealing and responsive frontend** using HTML5, CSS3, and JavaScript to display project cards, interactive elements, and smooth navigation.
- Utilized **Flask as the backend framework** to manage routing, handle form submissions, and dynamically render project content using Jinja templating.

## AUTOMATION AND SCRAPING PROJECTS

### Web Automation Capstone project

Tech Stack: Python, Selenium, BeautifulSoup

Link: [Web Automation](#)

- Developed an end-to-end automation tool that **scrapes structured data** from a target website and **automatically populates it into online forms**.
- Utilized **BeautifulSoup** to parse HTML content, extract relevant fields such as names, emails, or item listings from static web pages.
- Implemented **exception handling** and wait strategies (explicit and implicit waits) to manage page loading issues, broken elements, and slow responses.
- Handled complex form elements including drop-downs, checkboxes, and multi-step forms using XPath and CSS selectors.

### LinkedIn Job Apply Bot

Tech Stack Python, Selenium

Link: [LinkedIn Job Apply Bot](#)

- Automated job application process on LinkedIn using browser automation.
- Handled login, navigation to job pages, and form submissions.
- Logged each application attempt and ensured interaction reliability.

### Internet Speed and Twitter Bot

Tech Stack: Python, Speedtest CLI, Selenium, Tweepy

Links: [Internet Speed and Twitter Bot](#)

- Measured internet speed and tweeted complaints when below threshold.
- Used **Speedtest CLI** to fetch current speed stats – Scraped data from speedtest.ookla.
- Authenticated and posted tweets using **Tweepy** with stored API keys.

### Instagram Auto-Follower Bot

Tech Stack: Python, Selenium

Links: [Instagram Auto-Follower Bot](#)

- Built a bot to log into Instagram and automatically follow target users.
- Used **Selenium** to navigate through Instagram UI and simulate user actions.
- Incorporated delay and scroll mechanisms for human-like behavior.

### Amazon Price Tracker

Tech Stack: Python, BeautifulSoup, Requests, smtplib

Links: [Amazon Price Tracker](#)

- Built a bot using **Selenium** to automate browser interactions such as clicks and inputs.
- Implemented delay logic and error handling for dynamic elements.
- Used headless mode for performance testing and automation.