

## Flask

- Introduction to frameworks and Flask
- Flask Routes
- Database Layer in an application
- Database Layer Configuration
- Database Layer Schema
- Database Layer Relationships
- Database Layer CRUD
- Templating
- User Input
- Forms
- Form Validators
- Unit testing
- Flask Integration Testing
- Flask with Gunicorn
- Bcrypt

## Python Advanced

## Linux Intermediate

## CI/CD Basics

## CI/CD Intermediate

## NGINX

## Docker

## Docker Compose

## Docker Swarm

## Azure Introduction

## Azure Costs

## Azure Basics

## Azure Virtual Machines

## Azure Databases

## User Input

### Contents

- [Overview](#)
- [Forms](#)
- [CSRF Protection](#)
- [Tutorial](#)
- [Exercises](#)

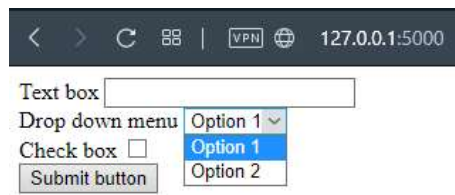
### Overview

Most web applications require some form of user interaction or input. This is implemented by using forms. The user interacts with the forms to send any required data to the app for any logic to be implemented.

### Forms

To create our forms for user input we are going to be using a Flask extension called WTForms.

While it is possible to implement forms using just HTML, WTForms offers a rich library of validators, fields, and CSRF protection support.

A screenshot of a web browser window. The address bar shows '127.0.0.1:5000'. Below the address bar, there is a form with four elements: a text box labeled 'Text box', a drop down menu labeled 'Drop down menu' with 'Option 1' selected, a check box labeled 'Check box' which is unchecked, and a button labeled 'Submit button'.


Above is an example of a text box, drop down menu, check box and a submit button created by WTForms.

### CSRF Protection

A Cross-Site Request Forgery (CSRF) attack is when an attacker tricks a web browser into executing an unwanted action.

A successful CSRF attack can lead to *unauthorised transfer of funds, changed passwords and data theft*.

Luckily, **Flask-WTF** has built-in protection which you can use, which requires only two things to set up.

- `form.hidden_tag()` - must be included in your front-end html on any page which contains `POST` method.
- `SECRET_KEY` - a variable defined in your back-end as an `app.config`, used to encrypt data during transit.  You can generate the key however you like, but to keep it secret it should be stored as an environment variable.

Once these two things are included, Flask-WTF uses them to provide your app with CSRF Protection.

### Tutorial

There is no tutorial for this module.

### Exercises

There are no exercises for this module.