

Web Forms & Input Validation

Friday, May 17, 2024 7:37 AM

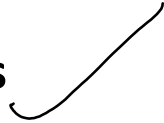
1. Forms



2. CSRF (Cross-Site Request Forgery)



3. wtforms



4. Code Demo

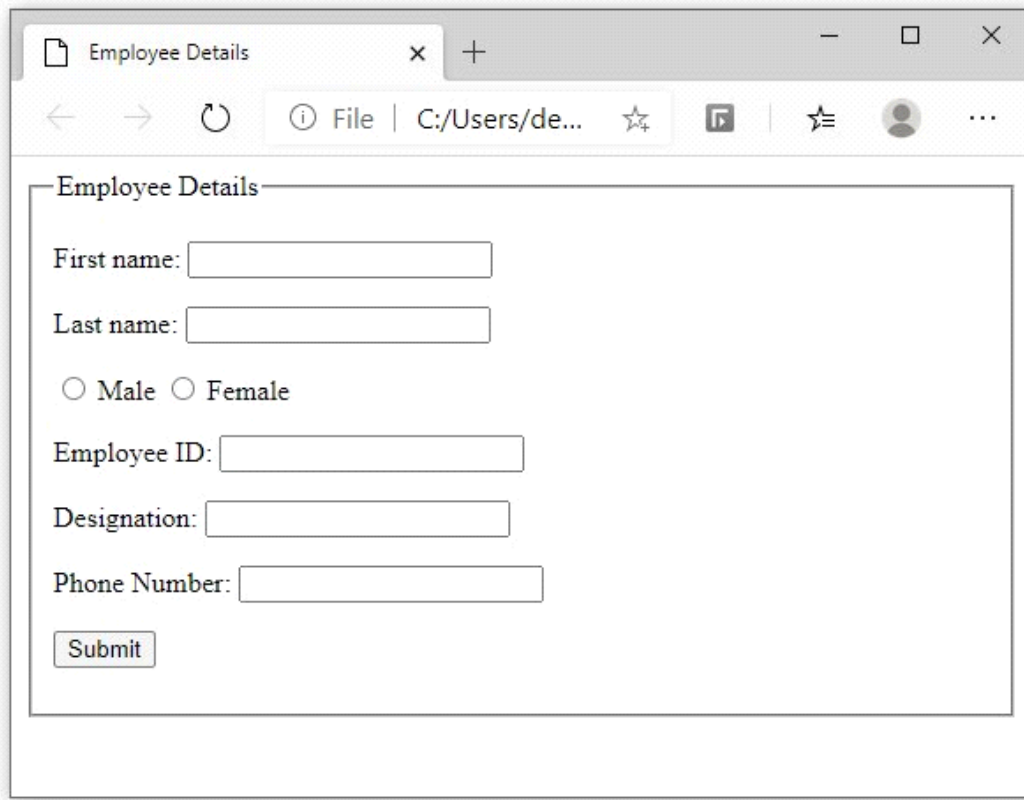


Forms

Friday, May 17, 2024 7:40 AM

1. What are Forms (Web Forms) ?

Forms are web pages where end-users can enter their data which can be sent to the back-end server for processing



The image shows a web browser window with a single tab titled 'Employee Details'. The address bar shows the file path 'C:/Users/de...'. The form itself is titled 'Employee Details' and contains the following fields: 'First name:' with a text input, 'Last name:' with a text input, gender selection with 'Male' and 'Female' radio buttons, 'Employee ID:' with a text input, 'Designation:' with a text input, and 'Phone Number:' with a text input. A 'Submit' button is located at the bottom left of the form area.

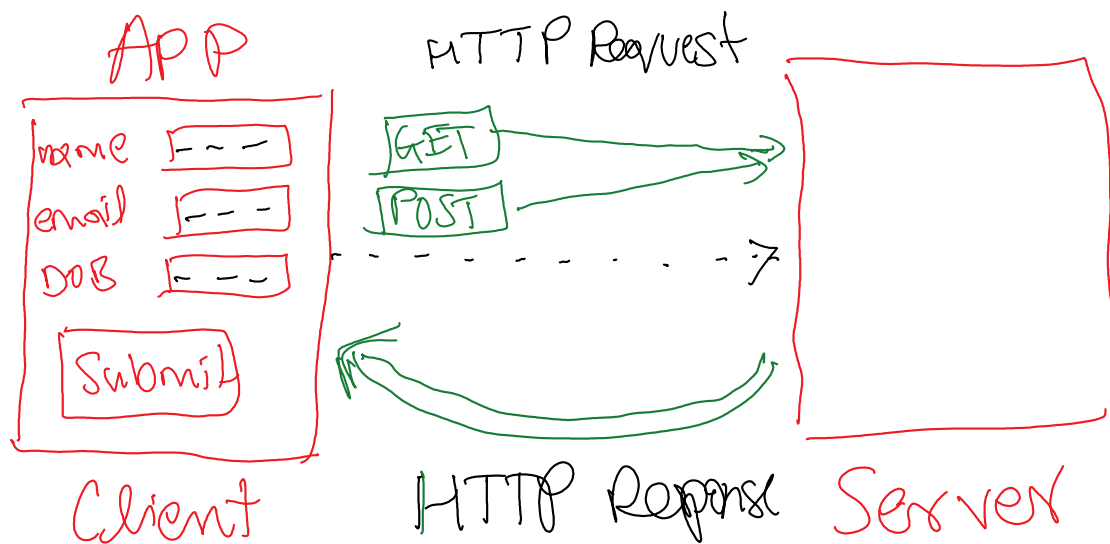
2. Why do we need Forms ?

- Allow end-users to interact with webpages and websites
- Enable data collection
- Facilitates communication with end-users
- Makes for dynamic websites

3. Working of Web Forms:

- Front-end design (HTML, CSS, JS)

- User input captured from input fields
- Input data processed in the back-end
- HTTP Request generated and sent to Server
 - GET - *query parameters*
 - POST - *store in database*
- HTTP Response sent from Server
- Response data parsed & displayed to Client accordingly



www.abc.com

Gender → Male } → www.abc.com? gender=Male &
 State → TX } state=TX

Name → . . .

Name
Email
DOB } → store DB

1. What is a CSRF attack?

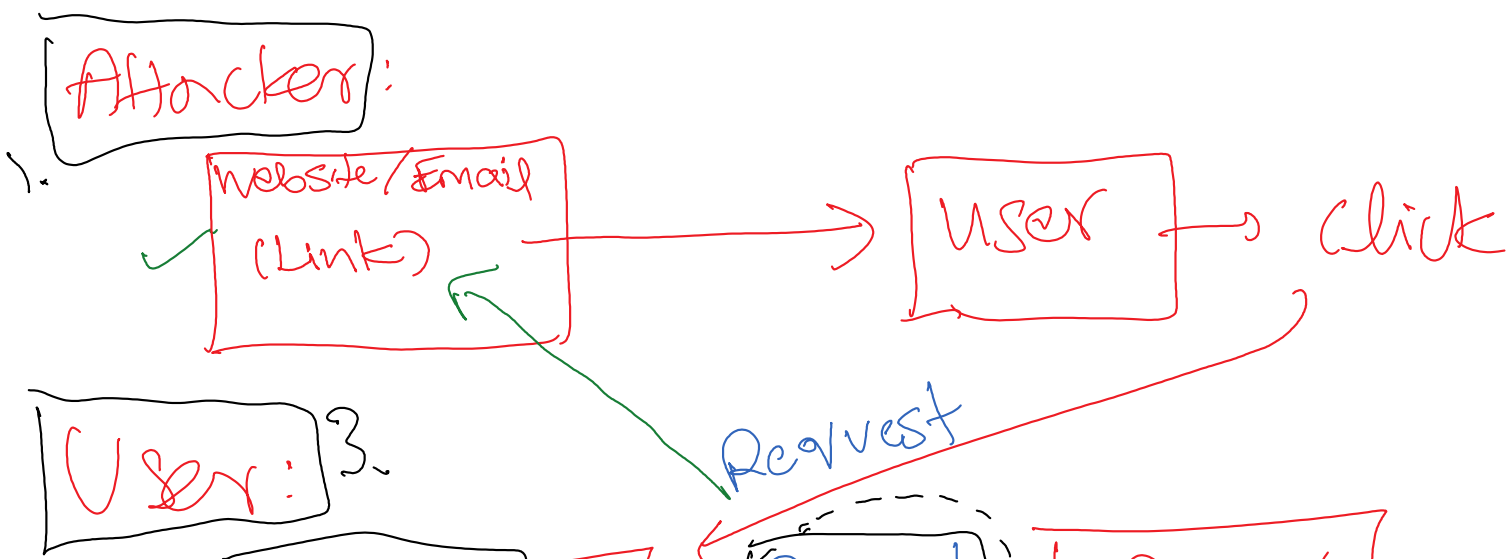
- It's a type of an attack, where an attacker 'tricks' an end-user in performing some malicious/unintended action over a web application
- Such attacks are common when web applications take inputs from users (ex: forms)

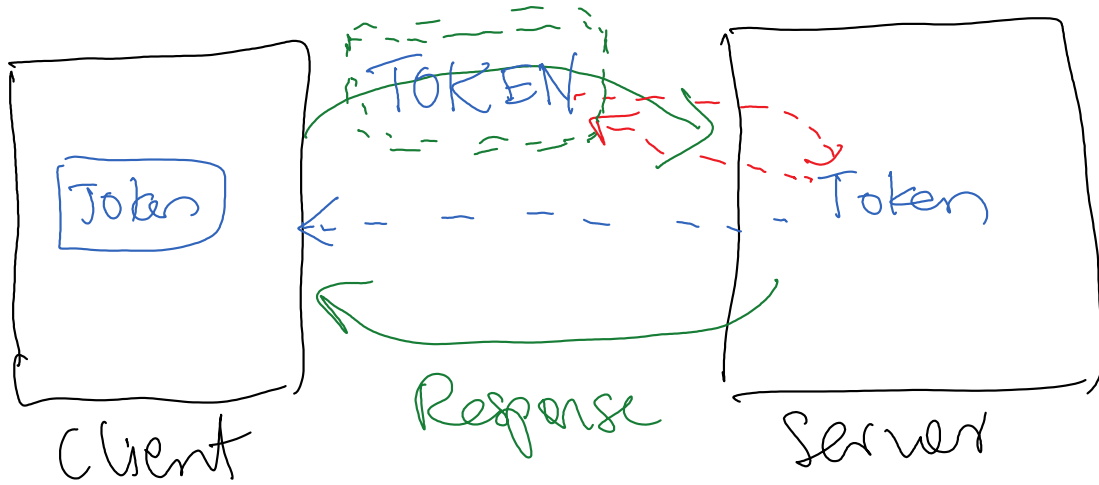
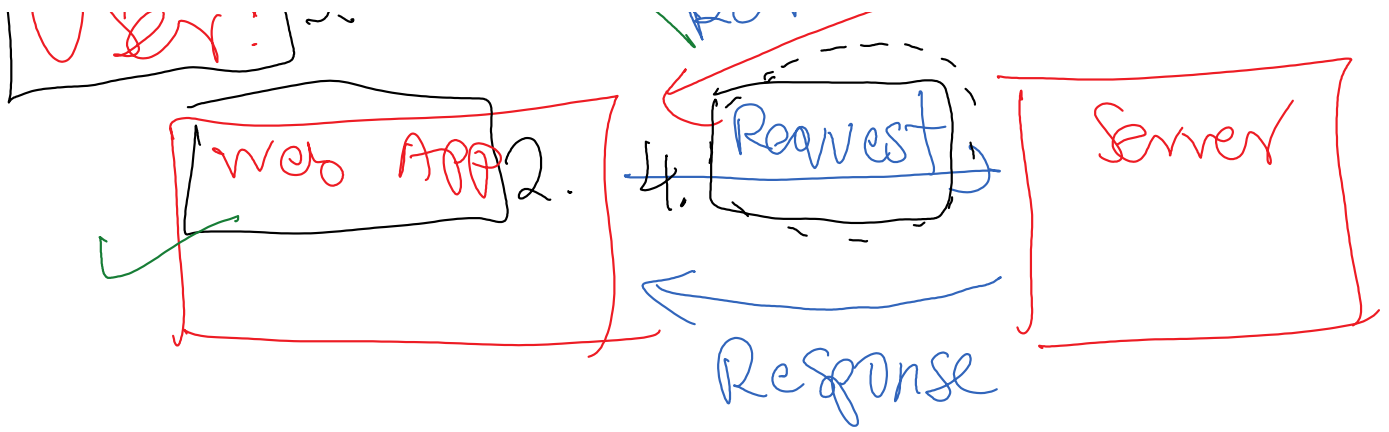
2. Working of CSRF:

- An attacker creates a malicious website or an email
- These websites/emails usually contain links that will generate a request to the target web application
- These requests are designed to perform some unintended actions over the target web application on behalf of the end-users
- Since the end-users would be authenticated, these 'unintended' actions would be perceived as 'genuine' by the web application and carried out
- These actions generally lead to leakage of sensitive data, transfer of funds, etc.

3. Prevention of CSRF: CSRF tokens

- It's a long, random and unpredictable string generated at the server side
- The token is shared with the end-user after authentication
- Whenever a request is sent to the server, the CSRF token must be provided
- The request is processed and a response sent, only when the token is validated





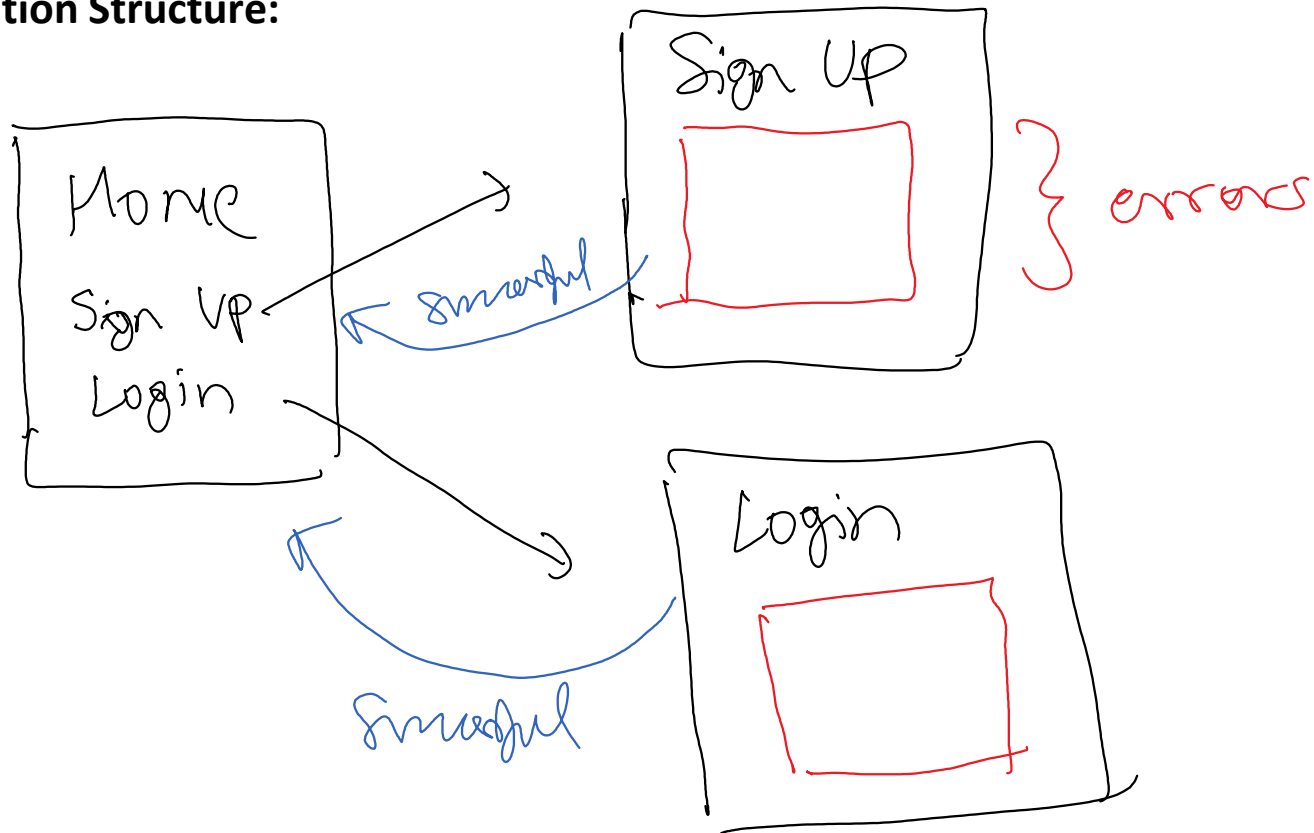
WTForms

WTForms is a flexible forms validation and rendering library for Python web development. It can work with whatever web framework and template engine you choose. It supports data validation, CSRF protection, internationalization (I18N), and more. There are various community libraries that provide closer integration with popular frameworks.

Why use wtforms ?

- Super convenient to create and handle forms
- Helps avoid manual input validation
- Helps prevent CSRF attacks
- Compliant with '*pythonic*' coding style
 - Treat forms as python classes
 - Work with forms as objects and attributes
 - No need to write tedious HTML code (input, label, type, id tags etc.)
- Integrated with Flask (*flask-wtf*)

Application Structure:



Sign Up

- username
- email
- gender
- DOB
- password
- confirm pw
- Submit

Login

- email
- pw
- remember me
- Submit

Submit

`<field_name>`

Submit