

FlaskWTF

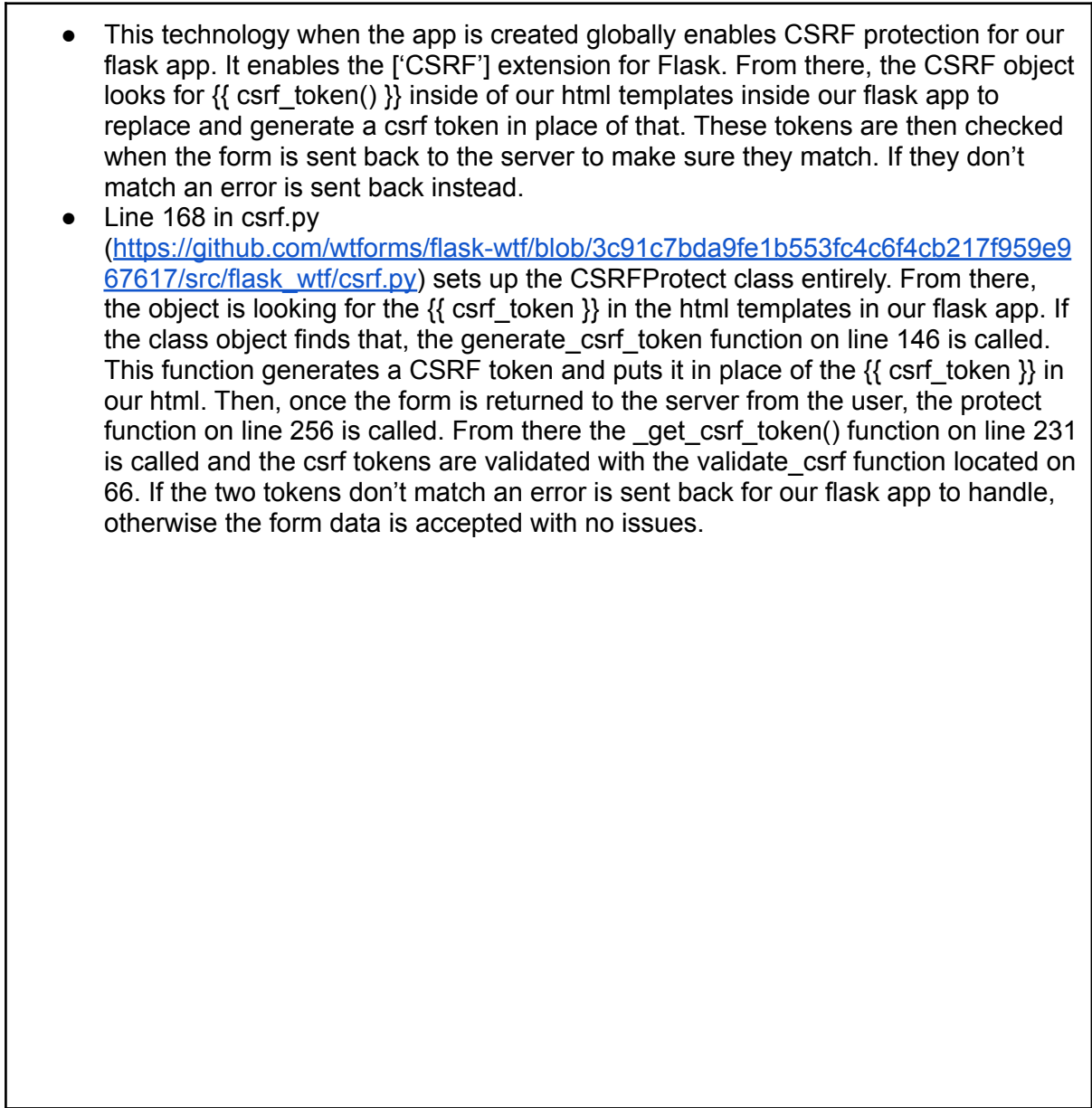
General Information & Licensing

Code Repository	https://github.com/wtforms/flask-wtf/
License Type	BSD-3 Clause License
License Description	Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met.
License Restrictions	<ul style="list-style-type: none">• Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.• Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.• Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
Who worked with this?	Collin

CSRFProtect Class

Purpose

- This class allows us to easily generate and validate CSRF tokens for our forms to add security to our web application.
- This tech is located on line 18 of app.py where our csrf object is defined. The CSRF tokens are generated when our html is created, which is handled by the CSRFProtect class. The tokens are generated in line 34 of login.html, line 34 of register.html and line 46 of settings.html



Purpose

- This tech allows us to handle a 400 Bad Request response if the CSRF data is tampered with before being sent back to the server.
- This tech is specifically used on line 54 of app.py with the `@app.errorhandler(CSRFError)`.

Dispel the magic of this technology. Replace this text with some that answers the following questions for the above tech:

- This technology does what is specified in our purpose section as Flask WTF directly integrates with our Flask app upon the initial setup. Because of this, we are able to use the CSRFError flag to handle CSRFError and send a 400 Bad Request Response if this is the case. We use a flask error handler here so that we can send back a custom html page for any tampered forms.
- The code that handles this is located in line 310 located in csrf.py (https://github.com/wtforms/flask-wtf/blob/3c91c7bda9fe1b553fc4c6f4cb217f959e967617/src/flask_wtf/csrf.py) accomplishes the task at hand with the CSRFError class at the bottom. This class object will be created whenever a BadRequest is received when an invalid CSRF token is received. The error_response function in line 306 creates a CSRFError class whenever it is called upon. The protect function located in line 256 checks all CSRF form data and will call the error_response function creating a CSRFError object whenever a form error is found. This will then call the error handler we have set up in our app.py in line 54.