

Week 5: Build Web apps with Flask: (Forms & Theming)

This week, we'll cover the following topics:

- Introducing Blueprints
- Design dashboard section (check figma UX designs for reference)
 - List all posts
 - Create Posts form
 - List all categories
 - Create categories form
- Flask Forms
- Form validations
- Introducing Jinja2 Macros
- Flash Messages ===== Requirement NOTE: To use flask forms, activate python environment with `pipenv shell` and run `pipenv install Flask-WTF`

Why Blueprints?

Blueprints in Flask are intended for these cases:

- Factor an application into a set of blueprints. This is ideal for larger applications; a project could instantiate an application object, initialize several extensions, and register a collection of blueprints.
- Register a blueprint on an application at a URL prefix and/or subdomain. Parameters in the URL prefix/subdomain become common view arguments (with defaults) across all view functions in the blueprint.
- Register a blueprint multiple times on an application with different URL rules.
- Provide template filters, static files, templates, and other utilities through blueprints. A blueprint does not have to implement applications or view functions.
- Register a blueprint on an application for any of these cases when initializing a Flask extension.

What Are Macros?

Macros are similar to functions in many programming languages. We use them to encapsulate logic used to perform repeatable actions. Macros can take arguments or be used without them.

Inside of macros we can use any of the Jinja features and constructs. Result of running macro is some text. You can essentially treat macro as one big evaluation statement that also allows parametrization.