

python/flask week 2 IP

[Start Assignment](#)

Due No due date **Points** 22 **Submitting** a text entry box or a website url

Goal

The goal of this project is to help us exercise and understand the concepts we have learnt during this week.

Overall Project Description

In context, your parents are busy workers and they get home really late. They always miss the news and they are very frustrated since they can't keep up with current affairs. You are a developer and will use your skills and prowess to remedy this problem. You will hence create an application that will help them list and preview news articles from various sources.

News API

The [News API \(https://newsapi.org/\)](https://newsapi.org/) provides a wonderful API that will allow you to complete your task. Read the documentation in order to fully access the features of the API, learn how to use it and know how to implement features that are required for your project.

User Story

1. As a user, I would like to see various news sources on the homepage of the application.
2. As a user, I would also want to select a news source and see all news articles from the selected news source in the application.
3. As a user, I would want to see the image, description and the time a news article was created.
4. As a user, I would want to click on an article and read the full article on the source website.

UI Design

Feel free to use any of the resources below or others apart from the ones listed to create a beautiful, intuitive, & user-friendly UI for your application.

- [Bootstrap \(https://getbootstrap.com/\)](https://getbootstrap.com/) - For responsive UI design.
- [Coolers \(https://coolers.co/\)](https://coolers.co/) - For selecting color schemes.
- [Google Fonts \(https://fonts.google.com/\)](https://fonts.google.com/) - For Beautiful fonts.
- [Wireframe \(https://wireframe.cc/\)](https://wireframe.cc/) - For mockup design.

Project Objectives

1. Your project should contain two models, one for the news source and one for the news article.

2. Your project must use views and routes.
3. Your project should consume the News API.
4. Your project must contain request(s) to the News API.
5. Your project should follow the proper folder structure.
6. Your application should be deployed to Heroku.

Technical Requirements

1. The project should have a specs markdown file that displays the project's specifications.
2. The model classes should have unittests to check for behaviour. The tests should pass.
3. The project needs to be deployed to Heroku.
4. The project should have clear commit messages.
5. The project should be polished in a portfolio-quality state.

Further Exploration

1. You can add a search functionality to your application that searches for articles according to the source name.
2. Learn how to use flask sessions and allow users not to lose the news article snippet on relaunch of the applications
3. Create a favourite sources section where a user can add their favourite news sources and store them in a browser cookie.

Python News Highlight

Criteria	Ratings				Pts
The application works as expected, functionally achieving the user stories provided.	4 Pts All user stories are achieved All the user stories provided for this project work on the application.	3 Pts Half user stories achieved. The application has more than half of the user stories working.	2 Pts Less Than Half User stories The application has less than half of the user stories working	0 Pts No user story achieved The application none of user stories working.	4 pts
Project objectives are met.	4 Pts All project objectives are achieved. The application achieves all the project objectives listed.	3 Pts Half objectives met The application meets more than half of the objectives listed.	2 Pts Less than half objectives achieved. The application has less than half of the objectives working.	0 Pts No objective is met. The application has none of the objectives met.	4 pts
Commits are made regularly with clear commit messages associated with them.	3 Pts 20 + commits in the project The project has more than 20 commits that have clear commit messages associated with them.		2 Pts Between 10 and 20 Github commits The project has 10 to 20 commits with clear commit messages.	1 Pts Less than 10 Github commits The project has less than 10 commits with clear commit messages.	3 pts
Project is deployed.	3 Pts Project is deployed and working. The project is deployed to Heroku AND it works while in deployment.		2 Pts Project deployed but not working. The project is deployed to Heroku BUT it does not work in deployment.	0 Pts Project not deployed The project is NOT deployed to Heroku.	3 pts

Criteria	Ratings			Pts
Unittest	3 Pts Implemented Unittests All classes contain unittests that test the behaviour of the class and they ALL pass.	2 Pts Partly implemented Unittests The project contains some partial tests that all pass or partially pass.	0 Pts No unittests Project has no unittests.	3 pts
README markdown file.	2 Pts Project contains a proper README file. The project has a well documented README file. The file clearly outlines the description of the project, has a live link to the application's website, is clean of typos, has clear setup instructions, outlines the technologies used, authors and contains license information.	1 Pts Default README Project contains a default README file.	0 Pts No README file. Project has no README file included.	2 pts
Portfolio Quality State	3 Pts Well polished and visually appealing application. The UI design of the application is visually appealing, easy to use and in a portfolio ready quality state.	2 Pts Average styling The application has an average design of the UI.	1 Pts Minimal Design The application has no styling at all to make it look visually appealing.	3 pts
Total points: 22				