Typhoid Web App

I will first go through the available options and there pros and cons to give you a better idea of why I made the choices and then walk you through the steps I will take to build an app.

## User Interface:

* Below are few of the available options and there pros and cons to build the user interface.
  + Gradio
  + StreamLIT
  + Dash
  + Django
  + Flask

### Advantages and Disadvantages of each approach:

### 1). Gradio:

* Several UI components which are optimized for ML models
* Easy to install using pip.
* Fewer lines of code to get the UI up and running. It also generates a shareable link
* Gradio can be used in simple python script or in Colab.
* Only used to run single function. If we want to add additional functionality like authentication and signup than there is no option available.

### 2). StreamLit:

* Wide range of common UI components.
* Support wide range of libraries like OPENCV, Latex
* Does not integrate with Jupyter notebook.
* It requires learning its own syntax.
* Data upload limit is 50Mb

### 3). Dash:

* Powerful UI interface library to build analytical website
* Integrates with python, R, Jupyter and Julia.
* Open Source version doesn’t have that many features.
* One has to write more code than StreamLit and Gradio.

### 4). Django and Flask:

* Both are complete web building frameworks therefore they require the knowledge of HTML, CSS and Javascript to build the UI from scratch.

I would use Dash to build the production ready app as dash has all the right tools to build an analytical site or dashboard and Gradio for demoing.

## Frameworks:-

* Below are few of the python frameworks that can be used to build the app
  + Flask
  + Django
  + Fastapi

### Advantages and Disadvantages of each approach:

### 1). Flask:

* Flexible, Minimalist, beginner friendly framework. Easy to transit to a web framework because of in-built support.
* Prone to security risk because of third party modules.
* Requests are processed in queue that takes longer time.

### 2). Django:

* Easy to add functionality to your website.
* Django rest framework is used to build api’s
* Enhanced security against Sql injections and cross-site attacks
* SqLALchemy is better than Django ORM

### 3). fastapi:

* Build on standard as JSON schema, AUTH2.0 and OPENAPI.
* Validates data types even in nested JSON request.
* It is used to make fast websites. Speed and Scalability are its main plus points.

I will choose FastAPI to host the AI models using the API because it builds the fastest websites.

## Data Solutions:-

* Below are few of the data acquiring solutions and each one has its own advantages and disadvantages
  + Files.upload()
  + Mount the google drive
  + Google cloud storage, AWS S3, etc for file storage and data
  + Directly plug your database to your Colab using a package.

### Advantages and Disadvantages of each approach:

### 1). Files.upload ():

* The easiest approach with just few lines of code you can get the data
* If the data is big it will take a while and once the notebook closes the data is gone and at each start up you have to load the data.

### 2). Google Drive:

* It does not require manual reloading when restarting the notebook
* Access management and security risk for companies

### 3). Cloud based file Storage:

* Authentication can be used.
* Improves both code and your architecture
* Loading data will definitely be slower than the google drive.

### 4). Connect a relational DB to your colab:

* If already using DBMS than connecting your db to your colab is the best solution.
* Impossible to use with unstructured data.

The use of the database depends on the functionality we want to induce in our app. Whether we want to save or retrieve the patient’s profile or just make an app that does image classification? If we do want this functionality, than I would recommend SQL ALCHEMY to connect a relational DB to Colab.

## Proposed Typhoid App Building Steps

1). The User interface will have the option to sign up, create profile, and to upload a picture of the blood smear along with that, it will have a questionnaire about the patient and finally, a ‘submit’ and ‘cancel’ button.

2). I will use Google Colab server to run the Jupyter notebook as Colab servers have some of the best hardware.

3). For UI I will use Gradio to demo the ML model. Once we agree on the model I will use Dash for the front end

4). The ML model and routes will be served using the FASTAPI

5). SQLALCHEMY will be used to connect the DB.