*Mushroom Classification Project*

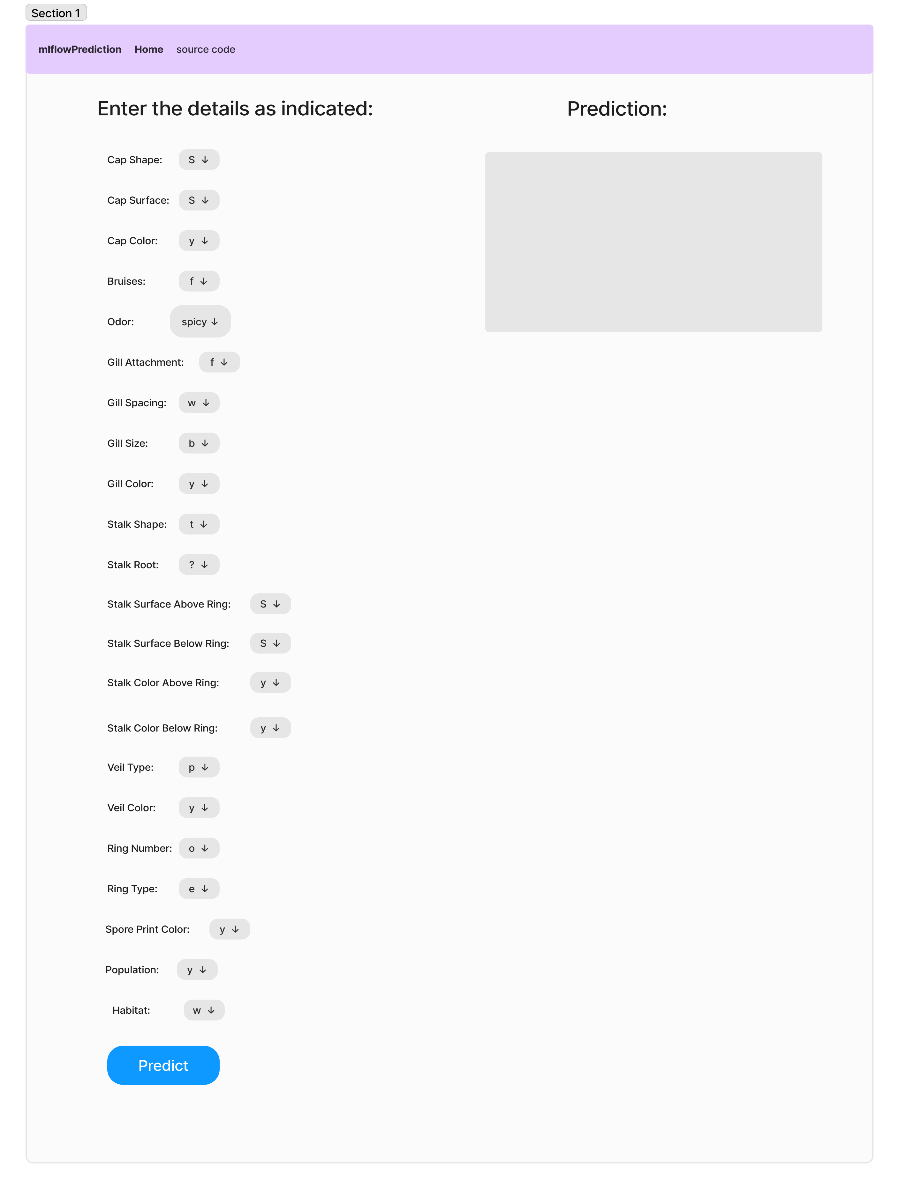
***Wireframe Documentation***



Homepage(localhost:3000)

We have designed a simple Wireframe of our Mushroom Classification project, It contains a Navbar which contains name of App, Home , Source Code (Github code url) on the left corner of navbar. Also Below it contains a simple form , Also to avoid any spelling mistake and wrong value error, we have provided a dropdown for each feature input .

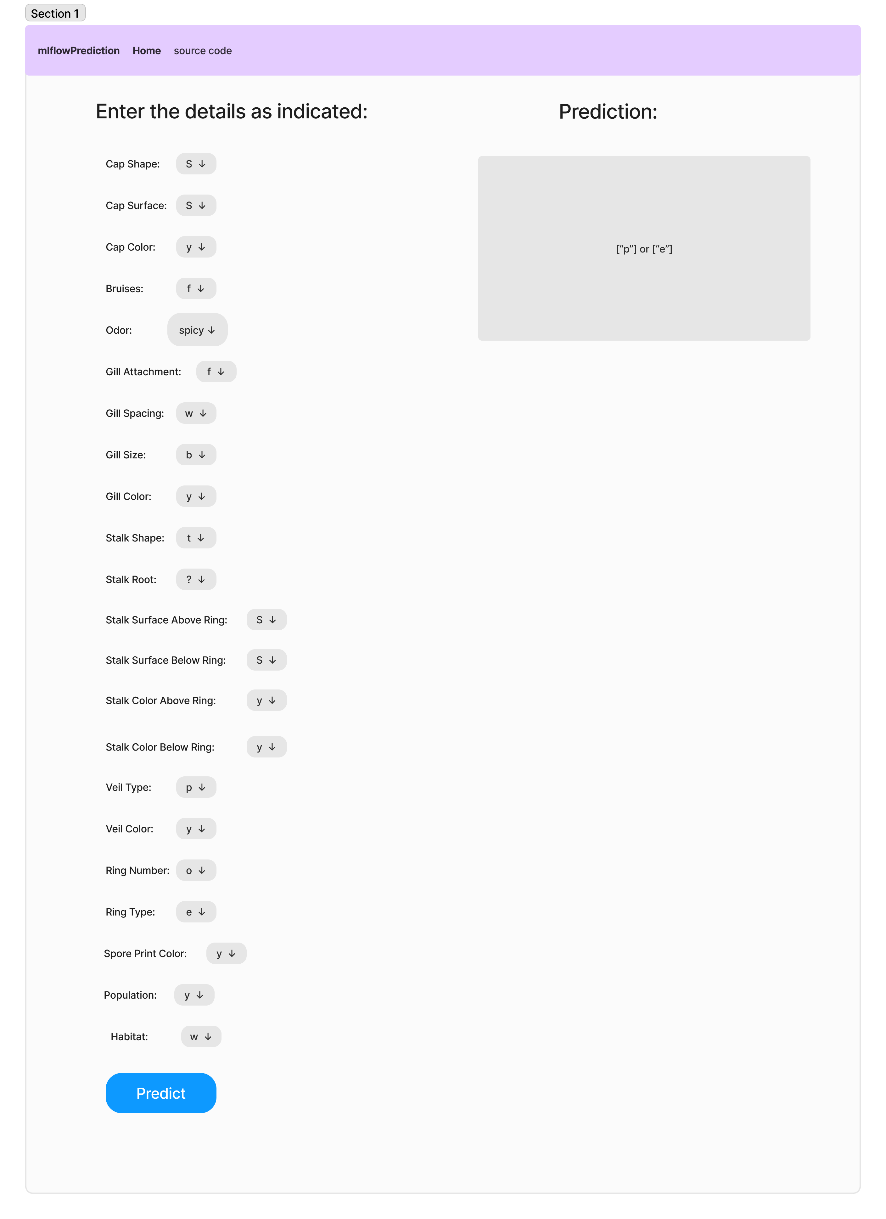
There is a empty Textarea which will contain the result of prediction after the user submits the form by selecting suitable values for each feature from the dropdown and clicking on ***predict*** button below the form.



On the Above , we can see the image/wireframe design of the Home Page of our web app . We can clearly see that it is a clean UI , simple to use and there is no complex design taken for this web app creation.

***Prediction Page***

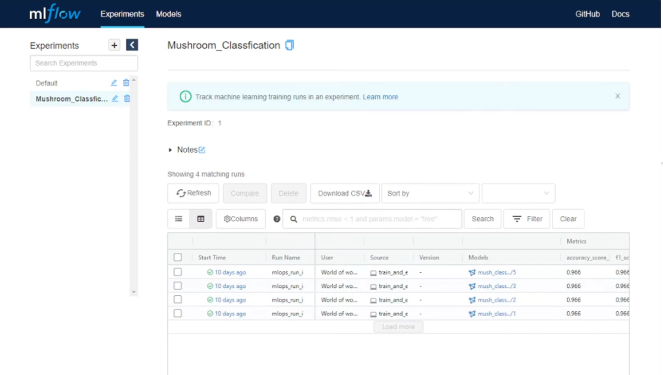
When the user clicks on predict button after selecting all the suitable values in the form of home page, The user gets a result from the model in the backend and displayed in the textarea below the Prediction . We can clearly see in the below image what our output will occur in the web page.



***Mlflow Page(localhost:5000)***

When we run docker-compose file , it launches two images on 2 ports (i.e, 3000 and 5000) on 3000, ther is flask app running and on 5000, we have mlflow tracking uri running. Here we can see all the experiments done by the Developer to build the models and also the logs of each model.

For each model, we have accuracy\_score,f1\_score,etc. Paramters are also logged like n\_estimators, learning\_rate, algorithm. By seeing these value of parameters and value of accuracy , f1, precision, We can compare which model is the best and at what parameters.



Also , We have ***Models*** button on the top of the Mlflow web page. When we go to models ,we get to see which model is in which state. For example :- Model 4 is in production state, model 3 is in dev stage and so on …

So we have automated this labelling of model as production , development , etc. And we will download and save production model in the Artifacts which will be further used for model prediction in flask app.

