# DATA GLACIER SIMPLE FLASK APPLICATION (DEPLOYMENT ON FLASK):

**Name: Blaise Papa**

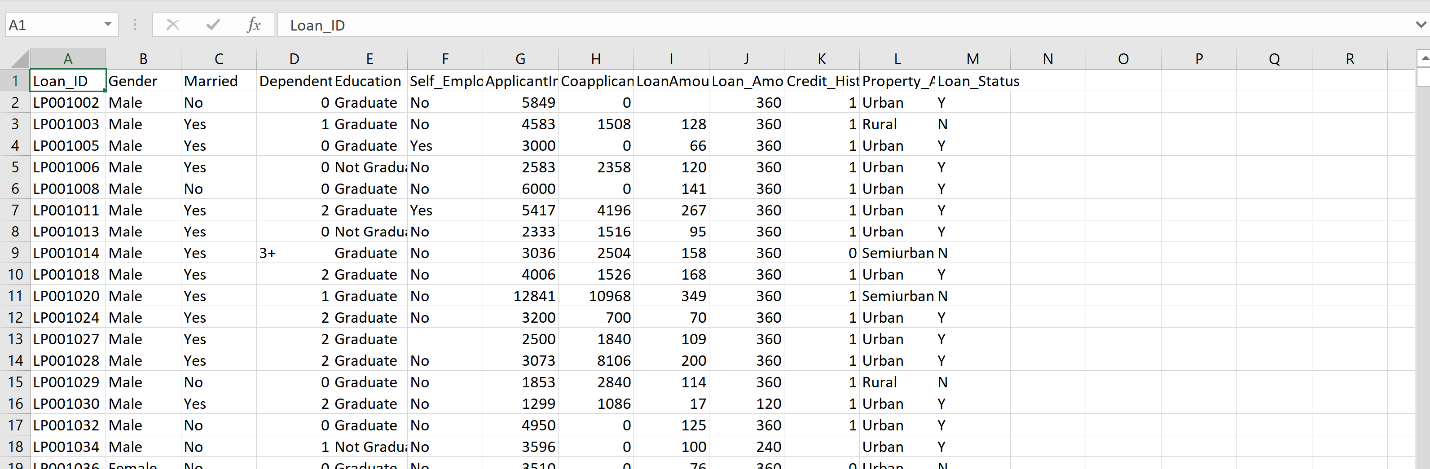
**Batch Code: LISUM01**

**Submission date: 3rd July 2021**

## GET TOY DATA AND LOAD

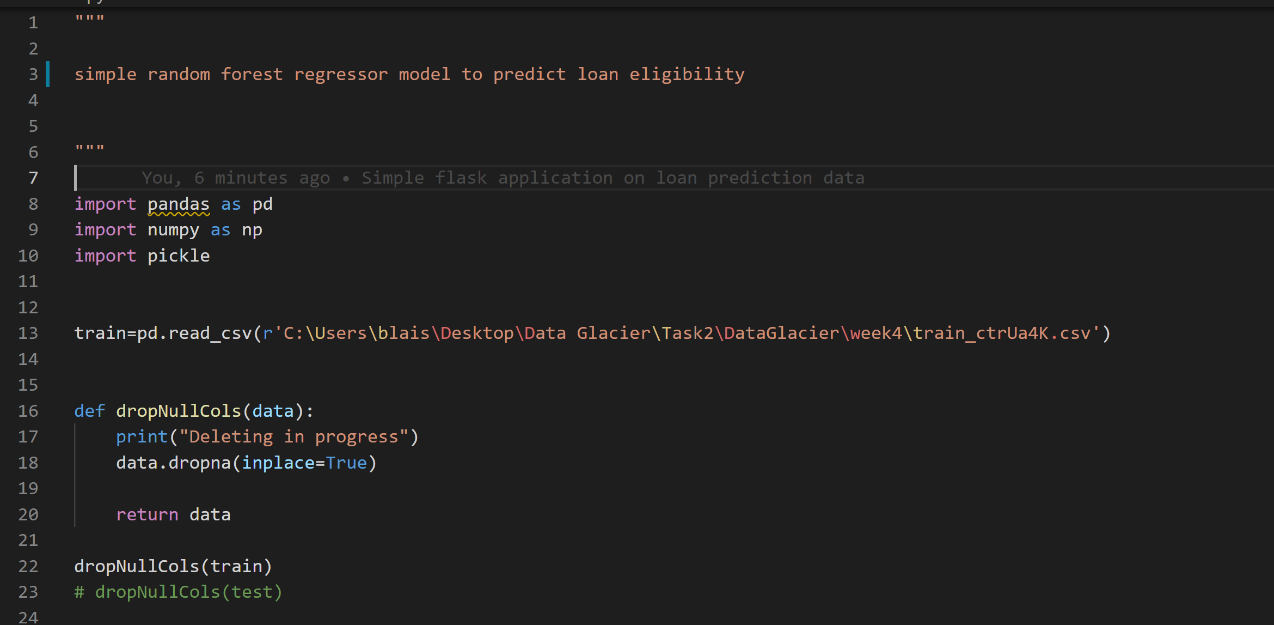
The model was developed around loan prediction data.

The data is collected on customer who apply for loans, the model is based to classify the customers between two classes; those whose loans are accepted and those whose loans are rejected.

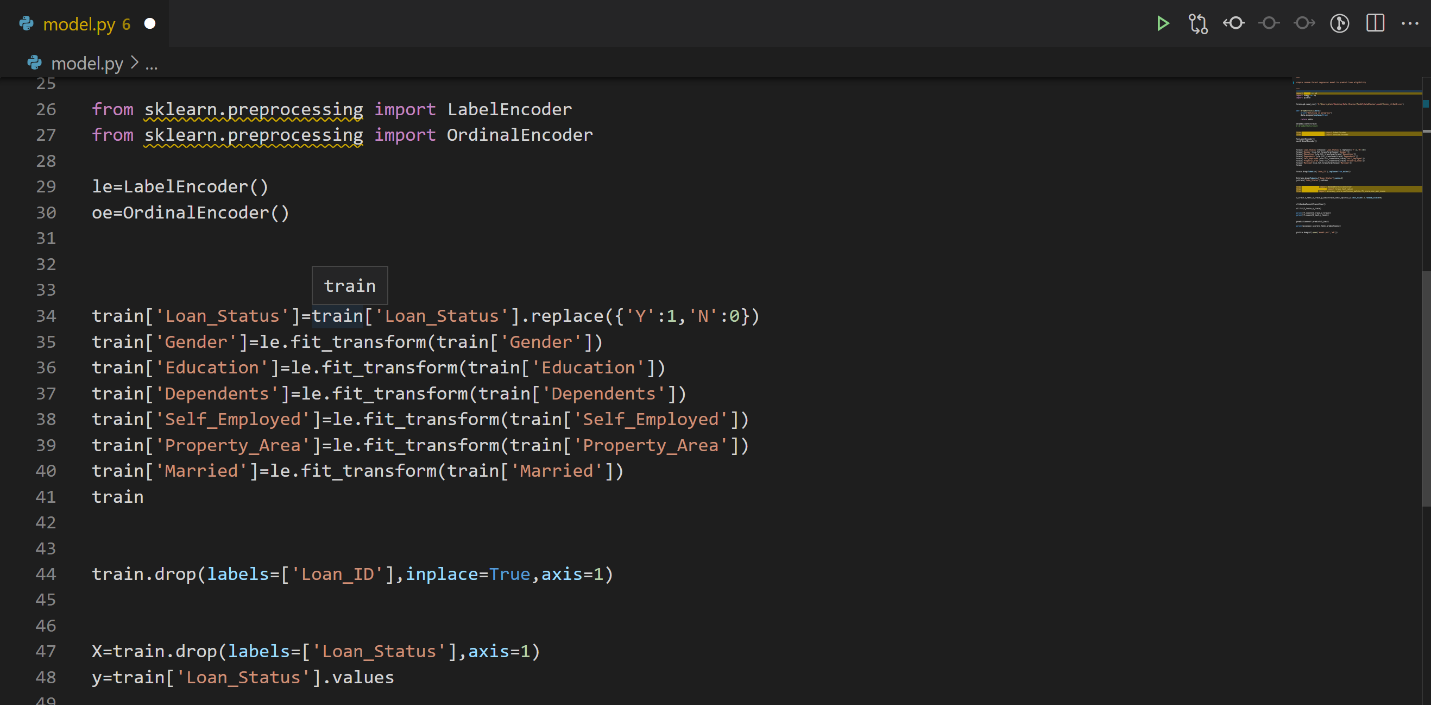


## MODEL BUILDING

The model is built and saved in the model.py file

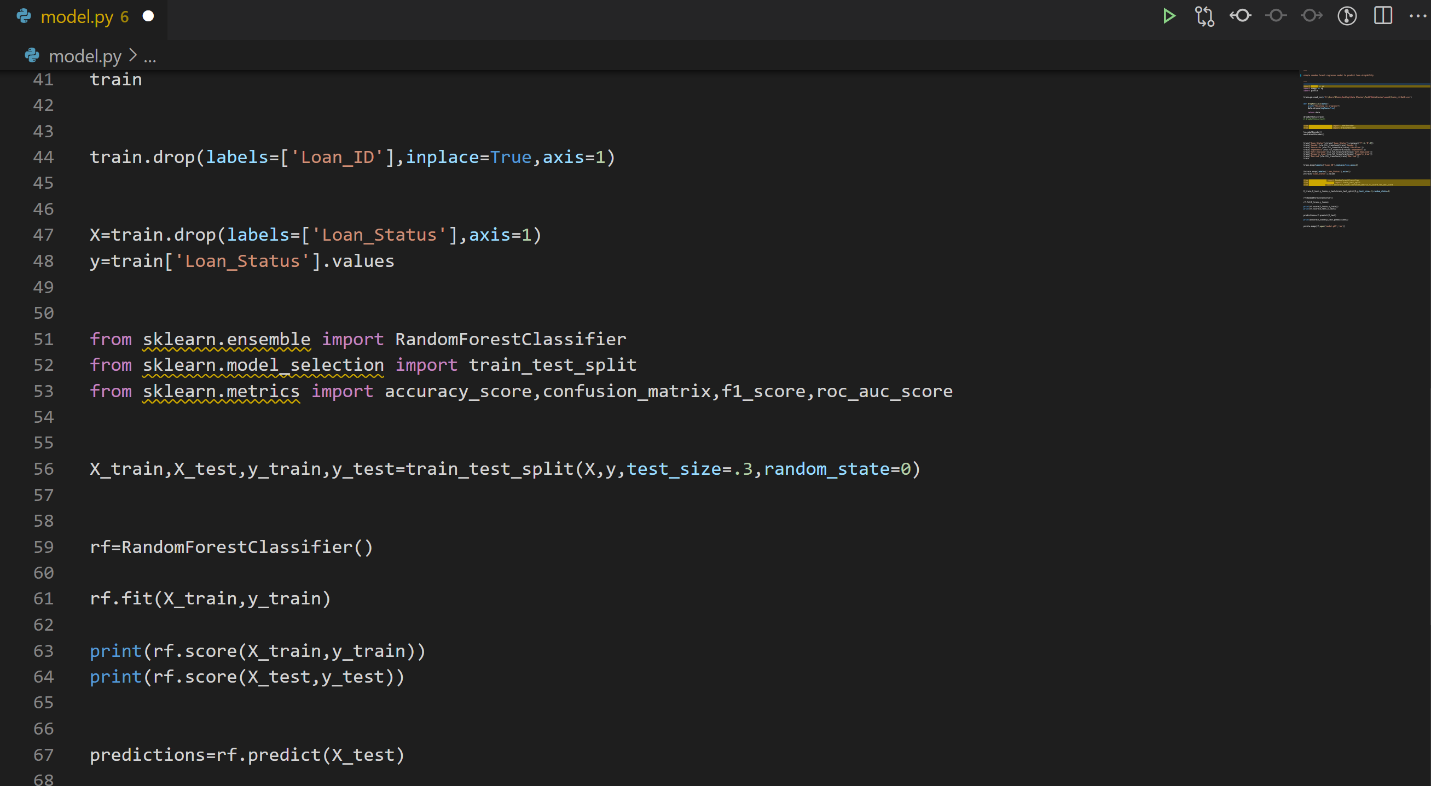
Loading data to python and converting into a data frame we can manipulate.

Simple data preprocessing steps to prepare data for model training. In this case we simply dropped all null column along with the ‘loan\_satus’ column and label encoded the object columns.



## MODEL TRAINING

For this binary classification model, we shall employ the random forest classifier, a well-known ensemble model. We also incorporate accuracy, f1 score and roc score as model evaluation metrics



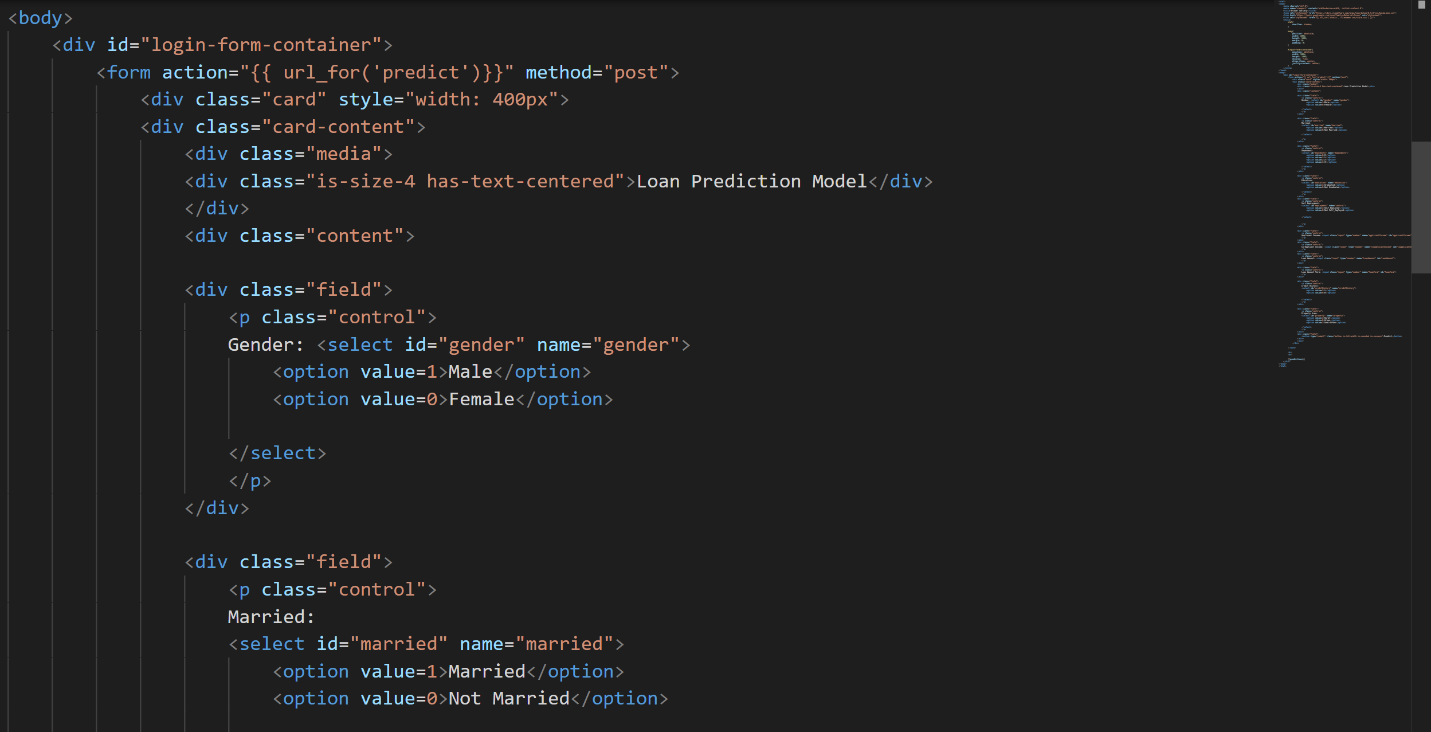
## MODEL SERIALIZATION.

Once satisfied with out model we serialize it into a pickle file which will enable us to deploy the trained model.



## BUILD SIMPLE HTML WEBSITE

We create a simple website that users are able to interact with.



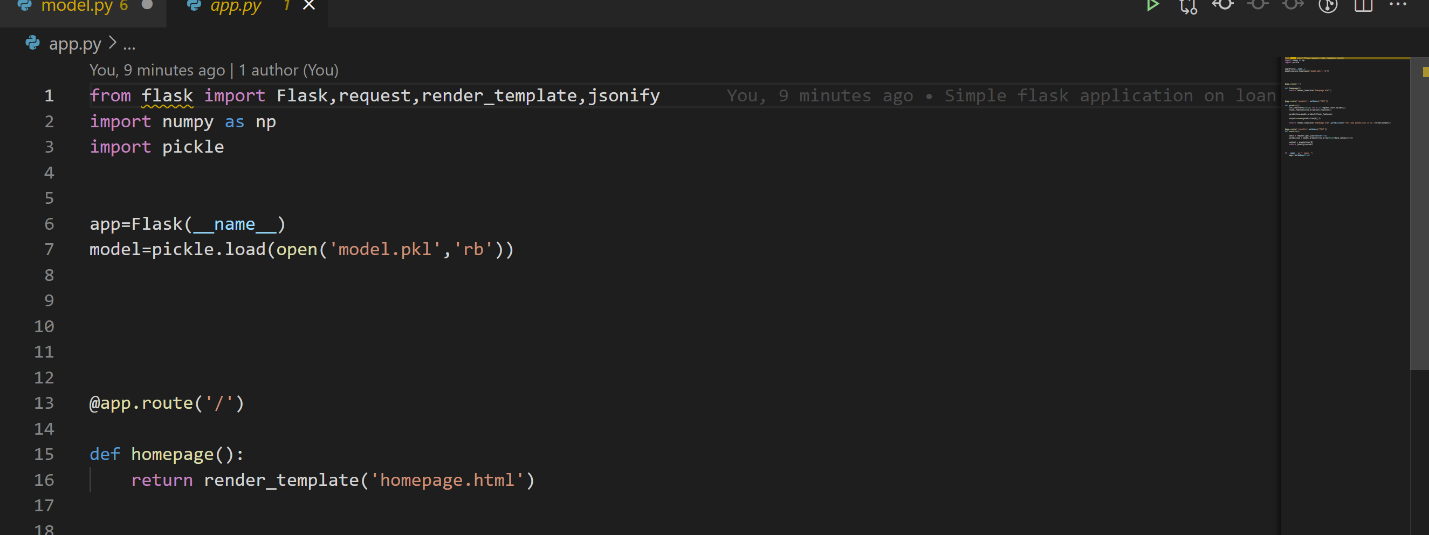
## BUILD FLASK APPLICATION

We create a simple flask application that will serve and deploy our model.

Model deserialization

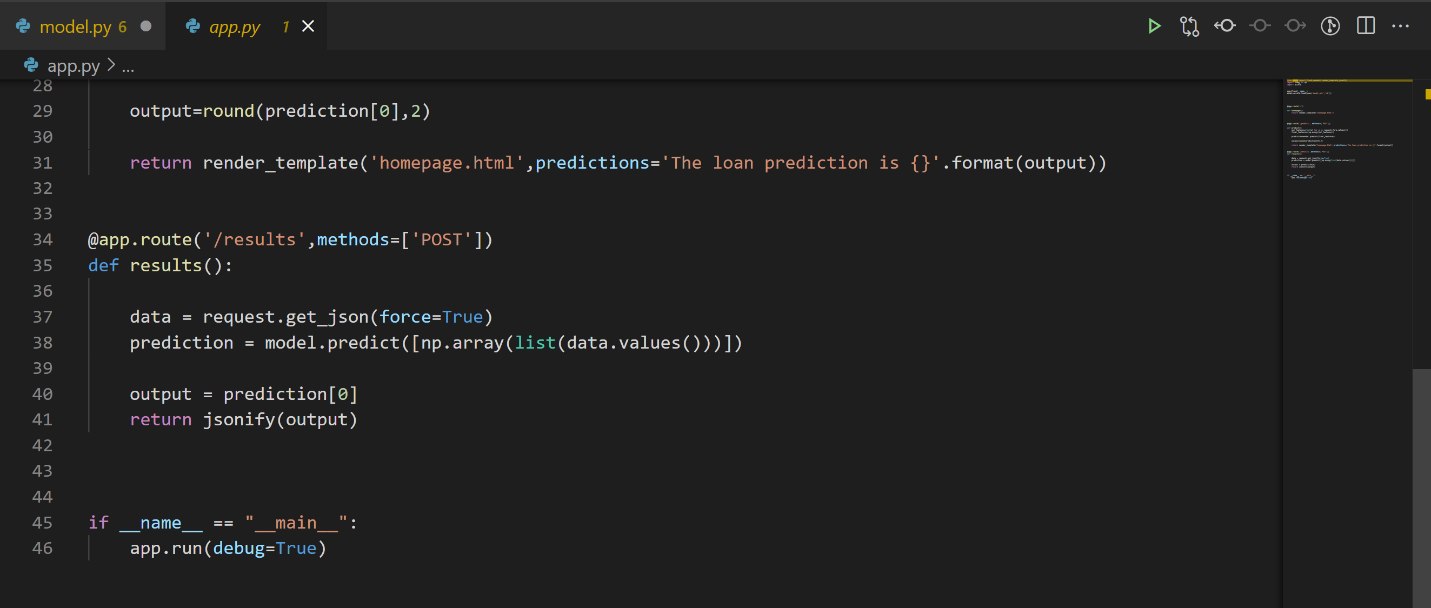
We define our flask application and then deserialize the model we earlier trained.

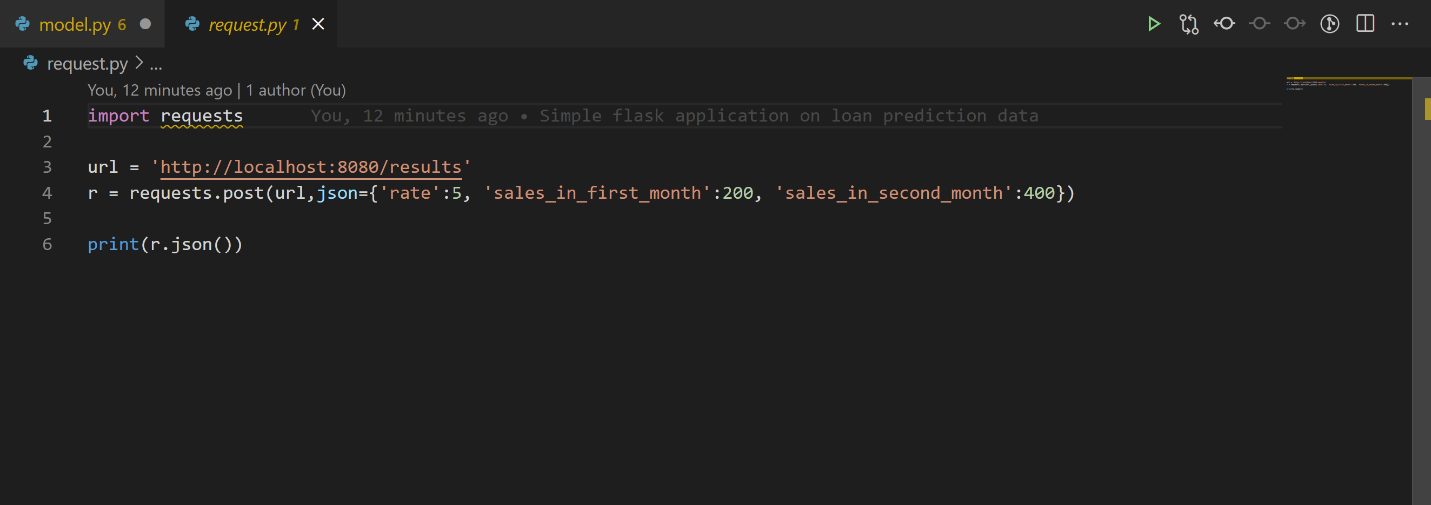
We create a default router which will be rendered when the default endpoint is called in this case the default endpoint(‘/’) renders the homepage.html file which is a simple website we had earlier created.



## BUILT ROUTES AND FUNCTIONS

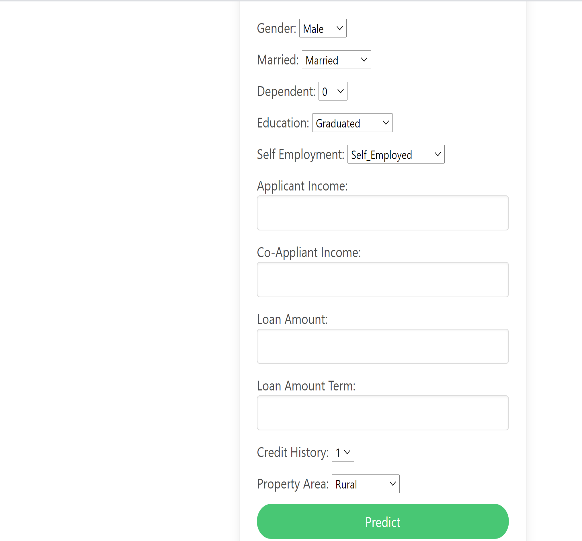
We define other routes that enable us to call various endpoints in our application



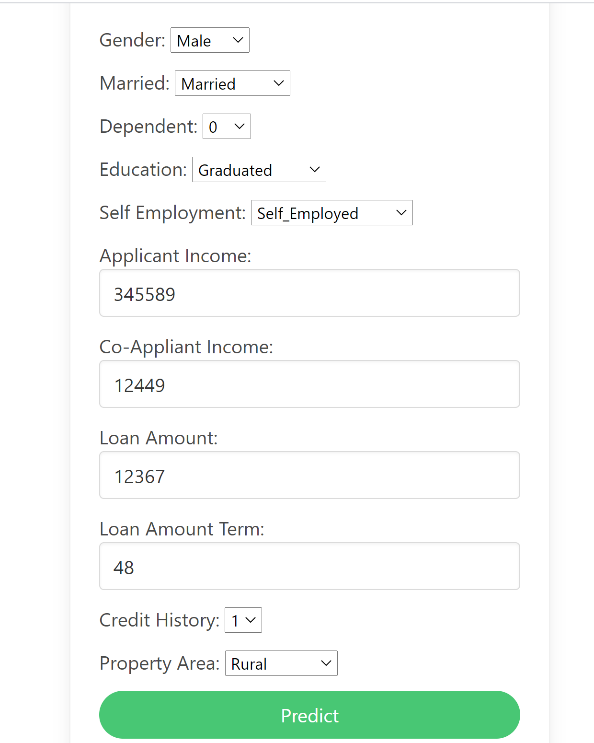


## DEPLOYED APPLICATION:

This is the rendered application.



The use is expected to fill in the form with details of the customer.



The model receives input and produces a prediction.

