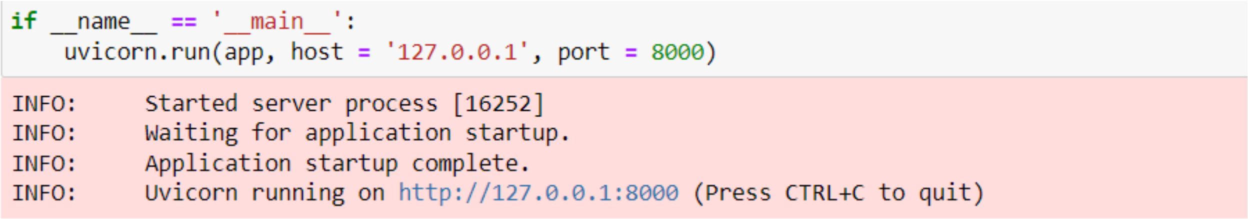
Wireframe Documentation

Flight Fare Prediction

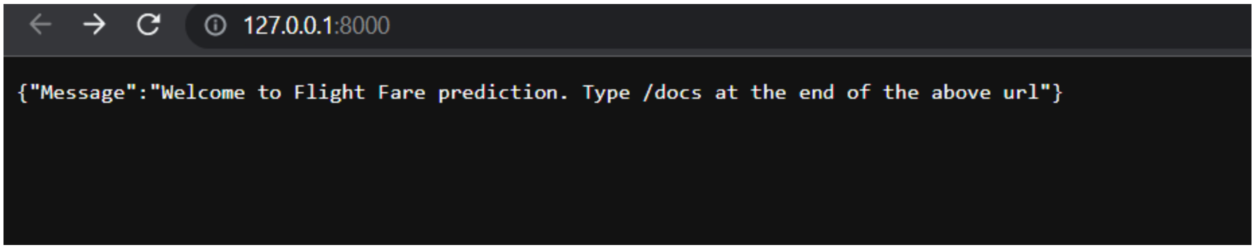
Home page

The model is deployed in FastAPI as an API and after running the link to the connection at 127.0.0.1:8000, the user is greeted with a home page as seen below



```
if __name__ == '__main__':
    uvicorn.run(app, host = '127.0.0.1', port = 8000)
```

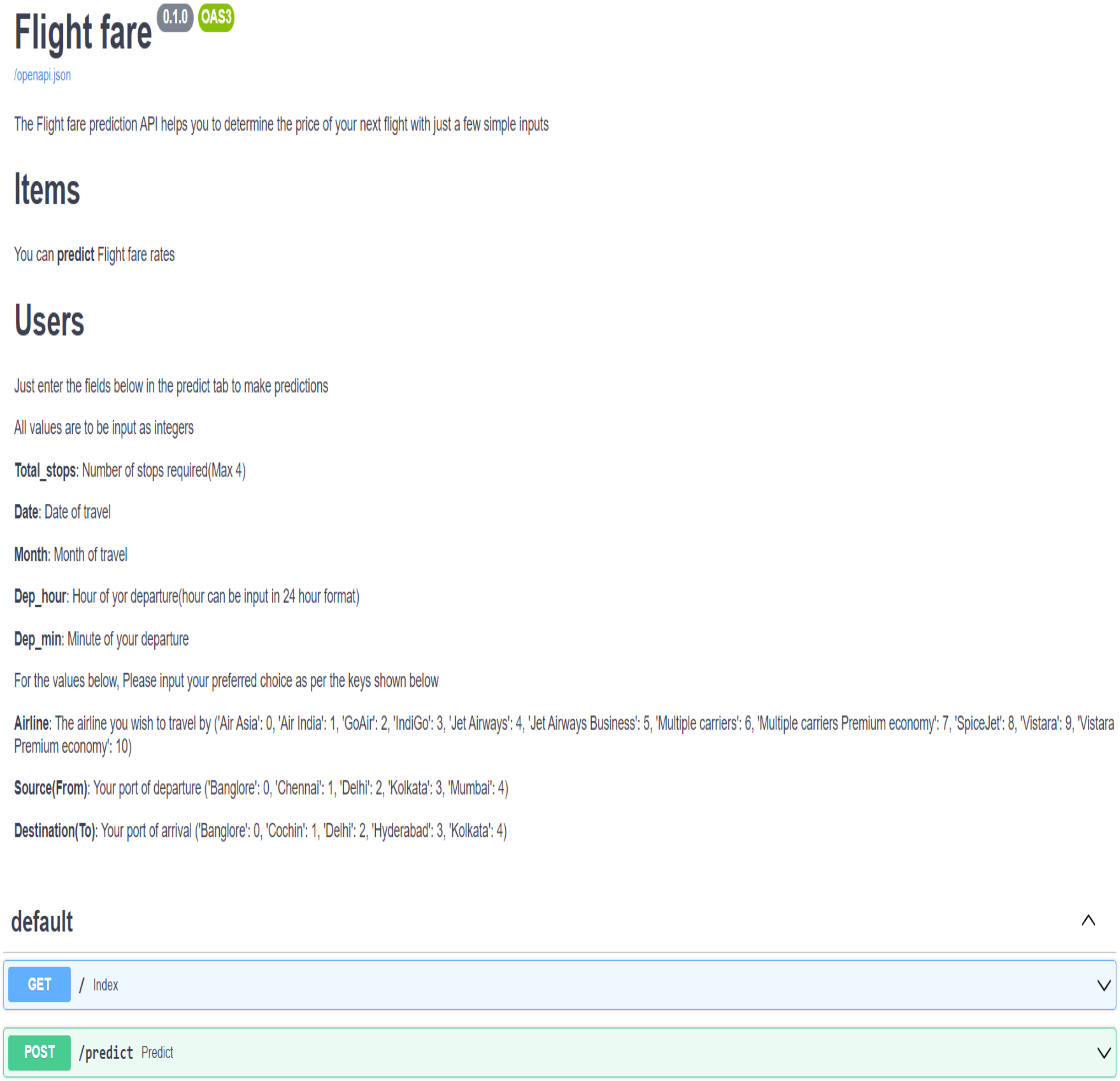
```
INFO:     Started server process [16252]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```



```
{"Message":"Welcome to Flight Fare prediction. Type /docs at the end of the above url"}
```

Prediction Page

The link can be diverted to the Swagger UI by typing “/docs” at the end of the link



Flight fare

/openapi.json

The Flight fare prediction API helps you to determine the price of your next flight with just a few simple inputs

Items

You can **predict** Flight fare rates

Users

Just enter the fields below in the predict tab to make predictions

All values are to be input as integers

Total_stops: Number of stops required(Max 4)

Date: Date of travel

Month: Month of travel

Dep_hour: Hour of yor departure(hour can be input in 24 hour format)

Dep_min: Minute of your departure

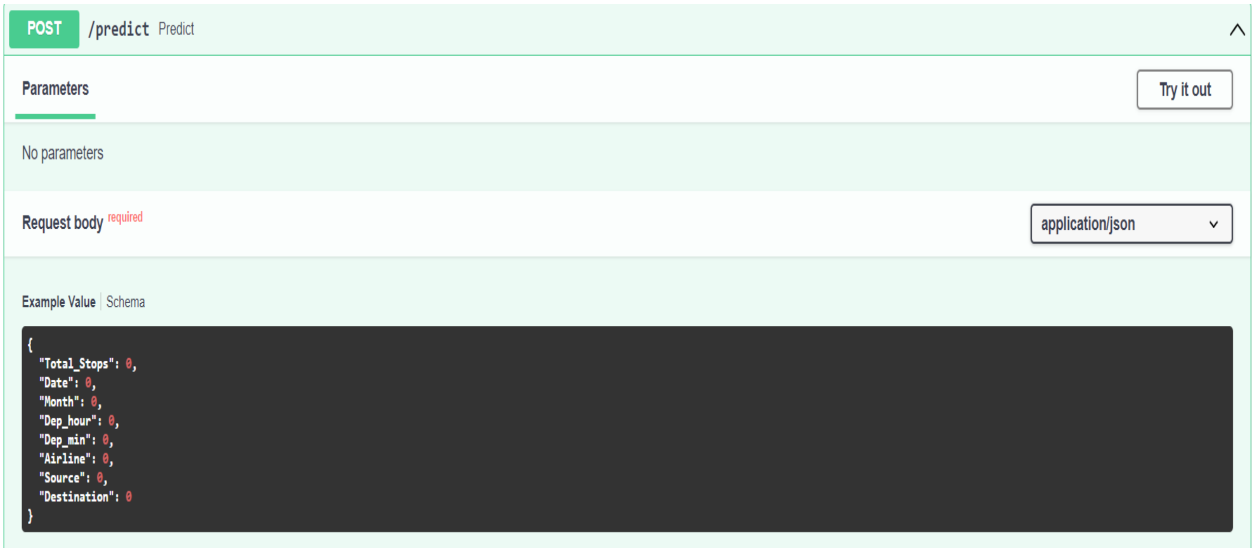
For the values below, Please input your preferred choice as per the keys shown below

Airline: The airline you wish to travel by ('Air Asia': 0, 'Air India': 1, 'GoAir': 2, 'IndiGo': 3, 'Jet Airways': 4, 'Jet Airways Business': 5, 'Multiple carriers': 6, 'Multiple carriers Premium economy': 7, 'SpiceJet': 8, 'Vistara': 9, 'Vistara Premium economy': 10)

Source(From): Your port of departure ('Banglore': 0, 'Chennai': 1, 'Delhi': 2, 'Kolkata': 3, 'Mumbai': 4)

Destination(To): Your port of arrival ('Banglore': 0, 'Cochin': 1, 'Delhi': 2, 'Hyderabad': 3, 'Kolkata': 4)

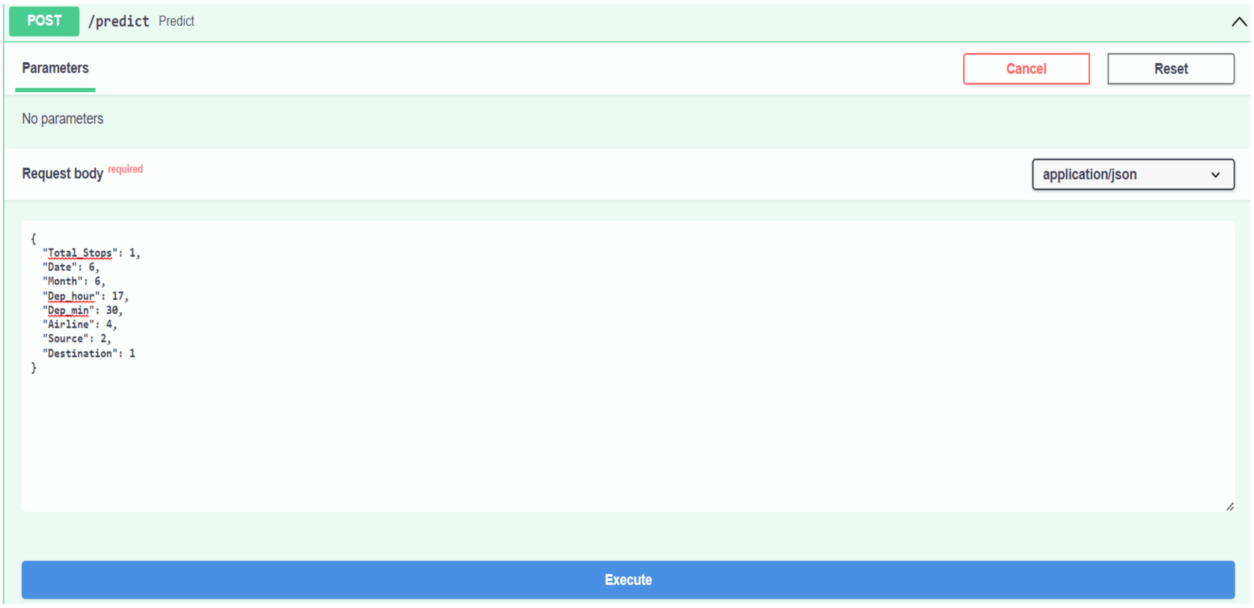
The user can start the prediction by selecting the predict pane and can input values so that the model can predict the fare of their trip.



No parameters

```
{
  "Total_Stops": 0,
  "Date": 0,
  "Month": 0,
  "Dep_hour": 0,
  "Dep_min": 0,
  "Airline": 0,
  "Source": 0,
  "Destination": 0
}
```

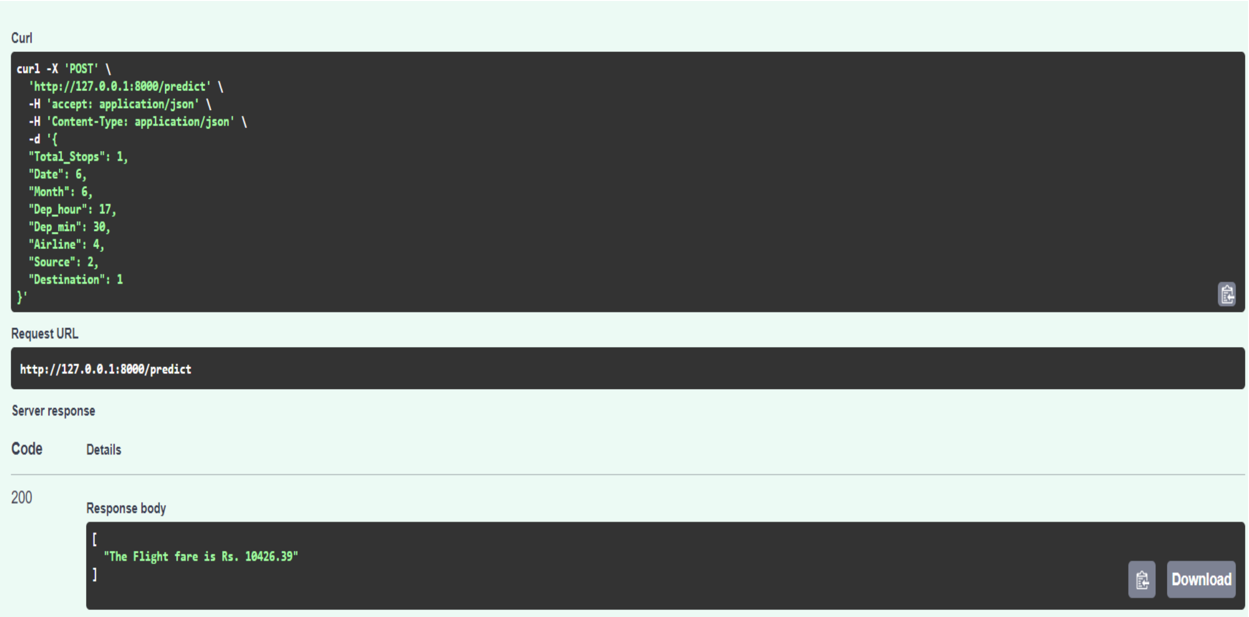
The values can be entered to the model by pressing on the “Try it out” button



No parameters

```
{
  "Total_Stops": 1,
  "Date": 6,
  "Month": 6,
  "Dep_hour": 17,
  "Dep_min": 30,
  "Airline": 4,
  "Source": 2,
  "Destination": 1
}
```

After pressing the “Execute” button, the model outputs the prediction below of the flight fare



Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Total_Stops": 1,
  "Date": 6,
  "Month": 6,
  "Dep_hour": 17,
  "Dep_min": 30,
  "Airline": 4,
  "Source": 2,
  "Destination": 1
}'
```

Request URL

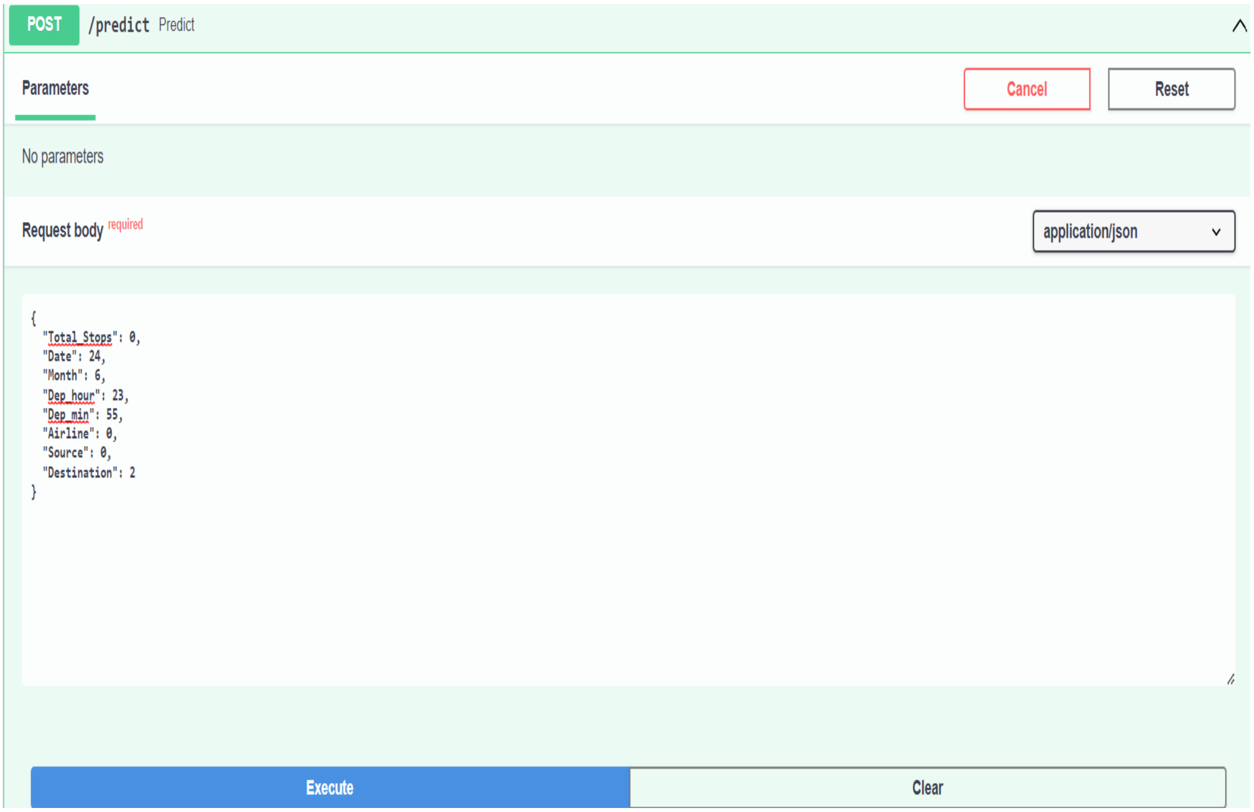
```
http://127.0.0.1:8000/predict
```

Server response

Code	Details
200	Response body

```
[
  "The Flight fare is Rs. 10426.39"
]
```

The model can be used to make predictions again by re–entering the relevant values as seen below



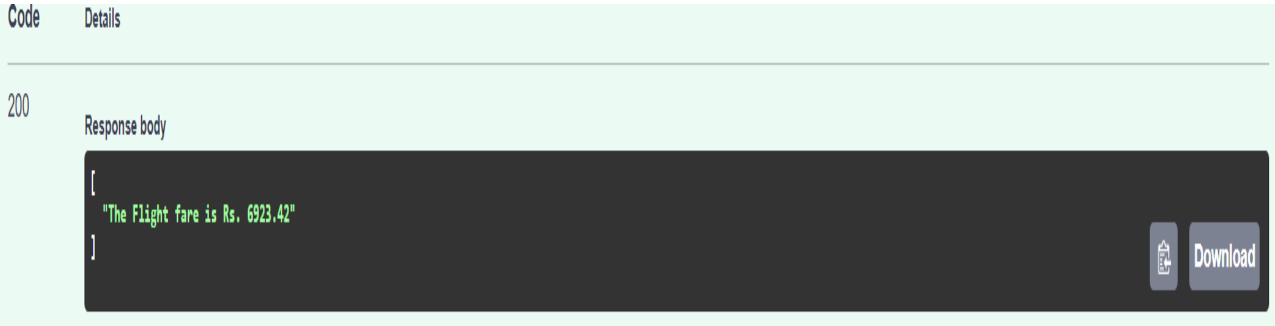
No parameters

Request body required

application/json

```
{
  "Total_Stops": 0,
  "Date": 24,
  "Month": 6,
  "Dep_hour": 23,
  "Dep_min": 55,
  "Airline": 0,
  "Source": 0,
  "Destination": 2
}
```

The model again outputs the prediction for the chosen flight.



```
[
  "The Flight fare is Rs. 6923.42"
]
```