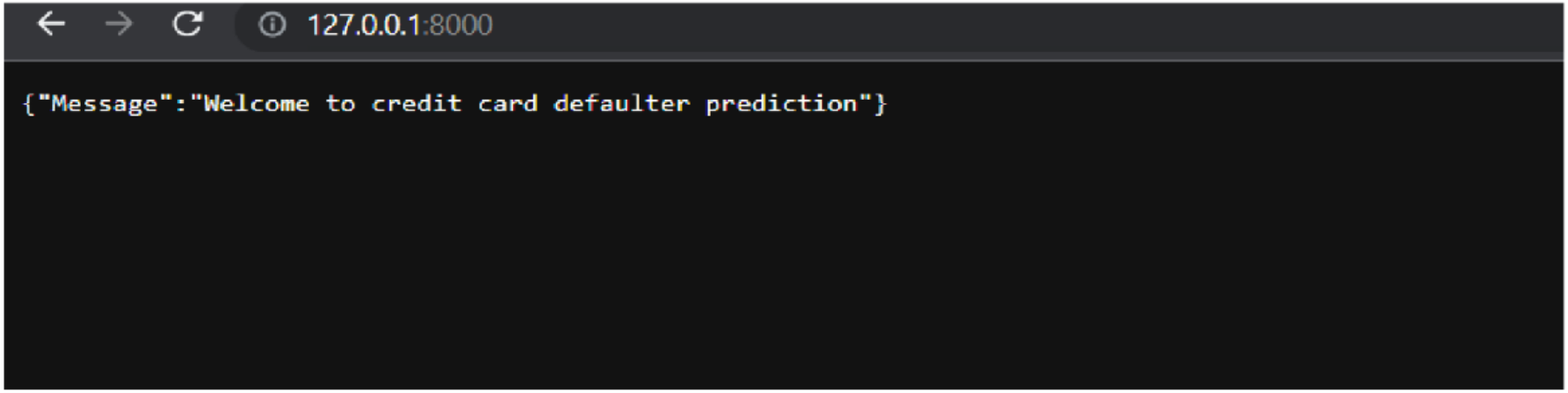
WIREFRAME

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

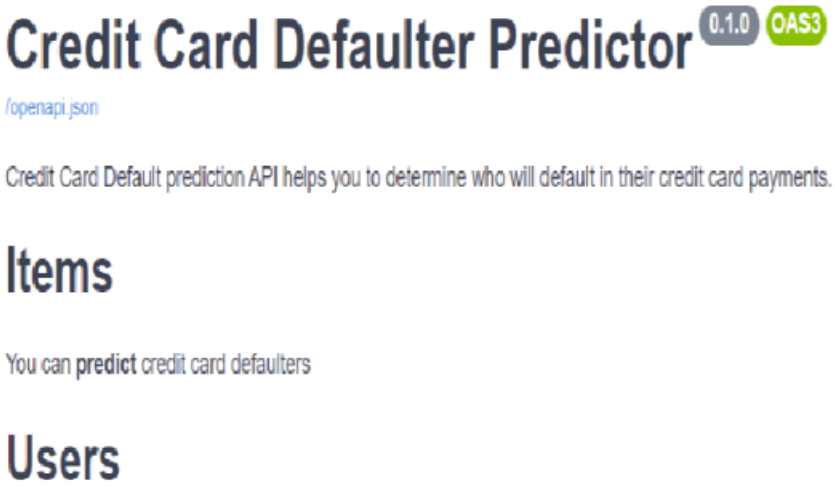
```
if __name__ == '__main__':
    uvicorn.run(app, host = '127.0.0.1', port = 8000)
```

```
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)
```


127.0.0.1:8000
{"Message":"Welcome to credit card defaulter prediction"}

PREDICTION PAGE

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



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

All values are to be input as integers

LIMIT_BAL: Amount of given credit in NT dollars

SEX: Gender (1 = Male, 2 = Female)

EDUCATION: (1 = Graduate School, 2 = University, 3 = High school, 4 = Others)

MARRIAGE: Marital status (1 = Married, 2 = Single, 3 = Others)

PAY_1: Repayment status in September (-1= pay duly, 1= Payment delay for one month, 2= Payment delay for two months, 3= Payment delay for three months, 4= Payment delay for four months, ... 8= Payment delay for eight months, 9= Payment delay for nine months and above)

PAY_2: Repayment status in August (same scale as above)

PAY_3: Repayment status in July (same scale as above)

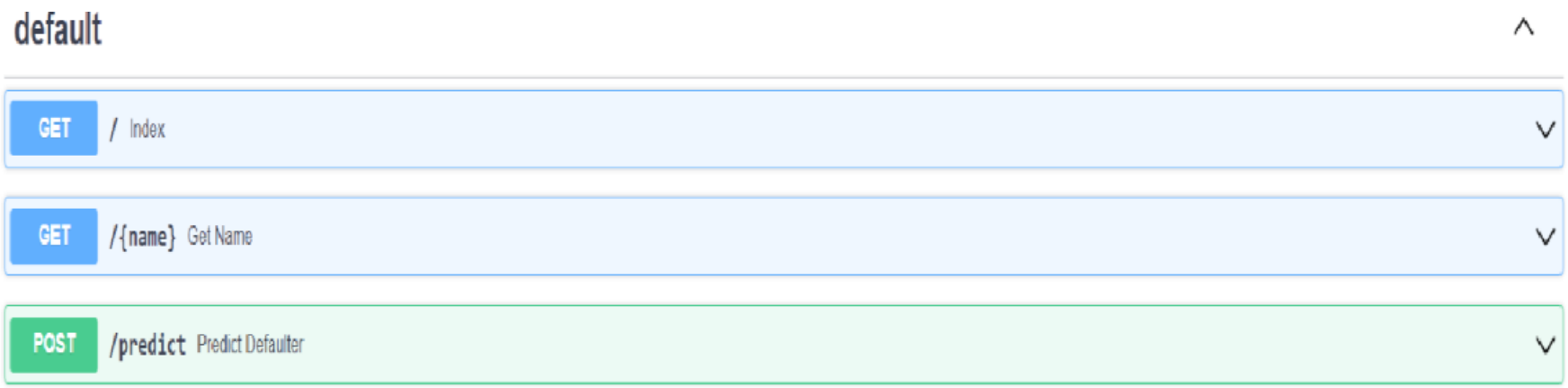
PAY_4: Repayment status in June (same scale as above)

PAY_5: Repayment status in May (same scale as above)

PAY_6: Repayment status in April (same scale as above)

Total bill amount : Total amount on bill for 6 months

Total paid amount : Total amount paid in 6 months



The user can start the prediction by selecting the predict pane and can input values so that the model can classify the customer as a defaulter or not.

POST /predict Predict Defaulter

Parameters | Try it out

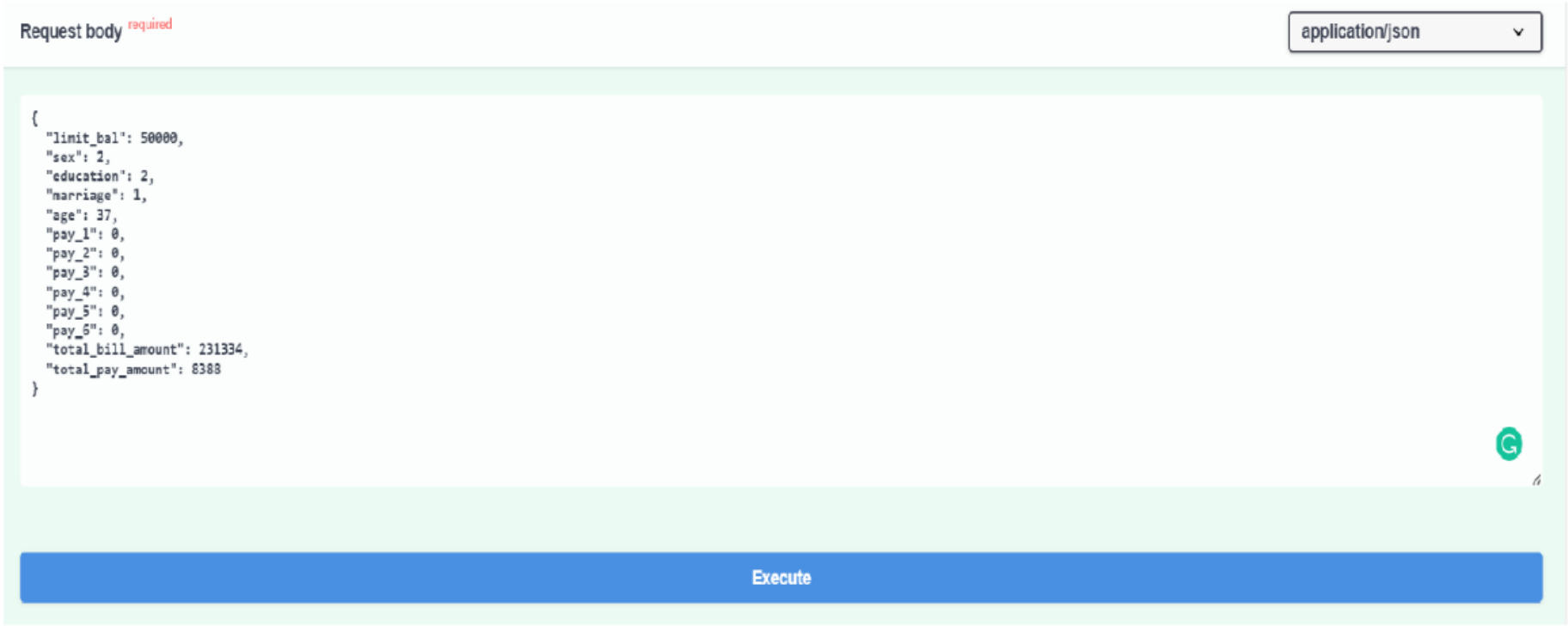
No parameters

Request body required | application/json

Example Value | Schema


```
{
  "limit_bal": 0,
  "sex": 0,
  "education": 0,
  "marriage": 0,
  "age": 0,
  "pay_1": 0,
  "pay_2": 0,
  "pay_3": 0,
  "pay_4": 0,
  "pay_5": 0,
  "pay_6": 0,
  "total_bill_amount": 0,
  "total_pay_amount": 0
}
```

The values can be entered to the model by pressing the try it out button.



After pressing the Execute button the model outputs the prediction below.

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "limit_bal": 50000,
  "sex": 2,
  "education": 2,
  "marriage": 1,
  "age": 37,
  "pay_1": 0,
  "pay_2": 0,
  "pay_3": 0,
  "pay_4": 0,
  "pay_5": 0,
  "pay_6": 0,
  "total_bill_amount": 231334,
  "total_pay_amount": 8388
}'
```

Request URL

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

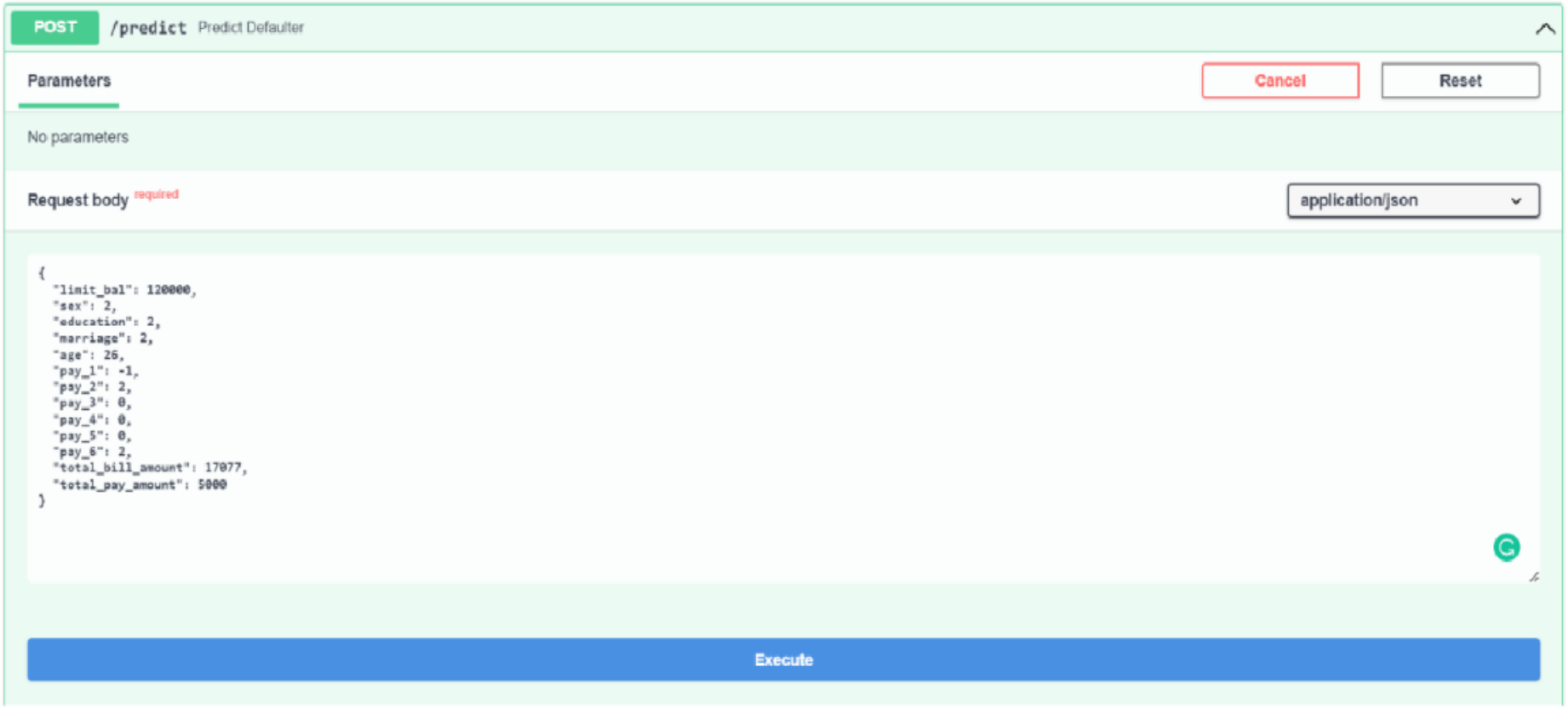
Server response

Code	Details
200	Response body

```
{
  "prediction": "Not defaulter"
}
```

Download

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



The model again outputs the prediction and it can be seen that the customer is a defaulter.

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/predict' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "limit_bal": 120000,
  "sex": 2,
  "education": 2,
  "marriage": 1,
  "age": 26,
  "pay_1": -1,
  "pay_2": 2,
  "pay_3": 0,
  "pay_4": 0,
  "pay_5": 0,
  "pay_6": 2,
  "total_bill_amount": 17077,
  "total_paid_amount": 5000
}'
```

Request URL

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

Server response

Code	Details
200	Response body

```
{
  "prediction": "Defaulter"
}
```

Download