BDL: A06

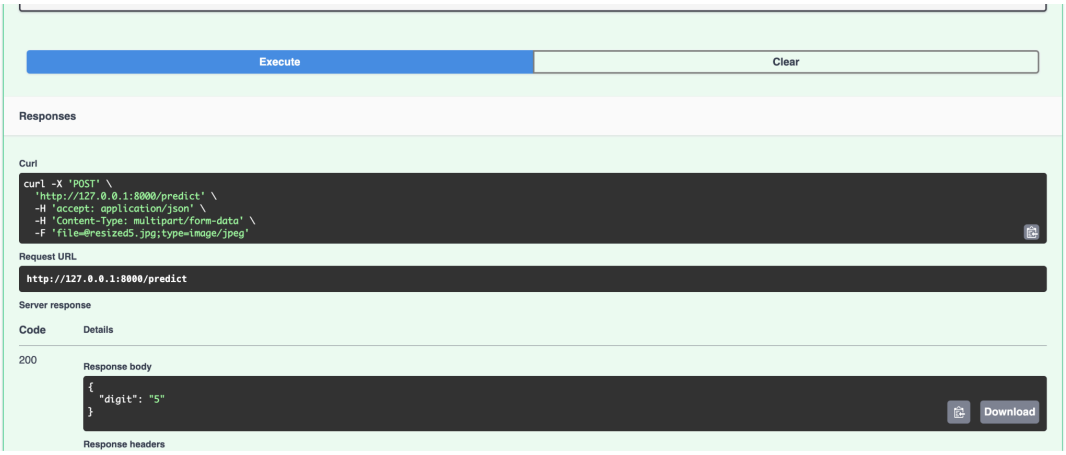
Nikhil Anand BE20B022

Task 1:

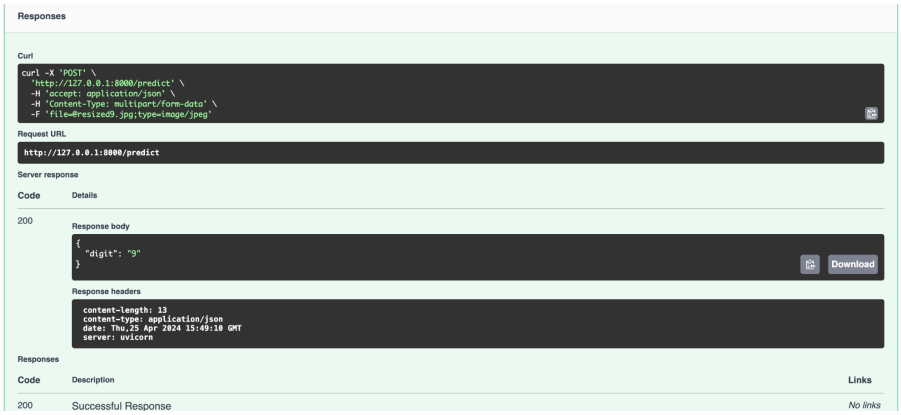
The following 3 images were created which are 28x28 images in the specific format that our model takes as input.

Thus, we may input this into the task1.py model server and get the predictions. (Without reformatting images)

For eg, when the 5 is inputted:



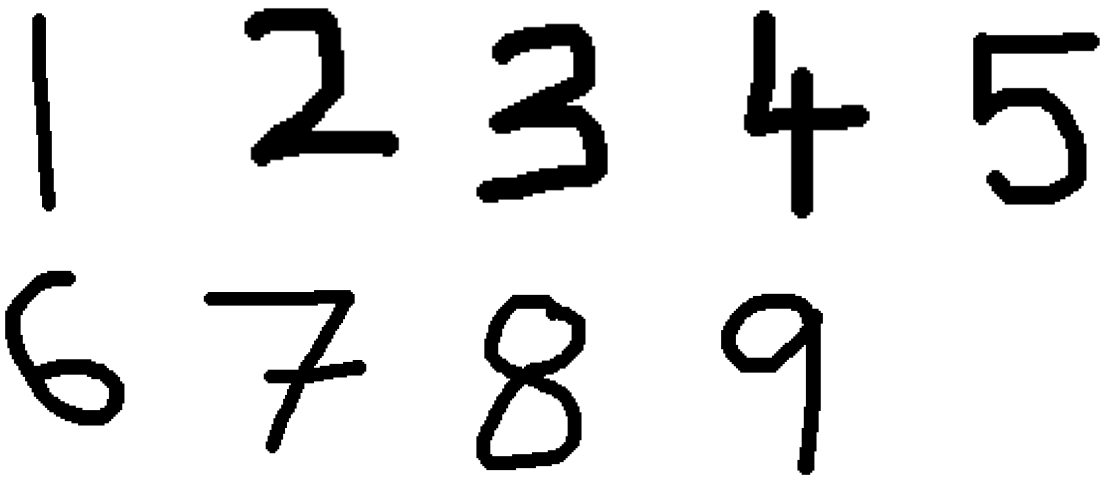
When the 9 is inputted:



Task 2:

The code has been implemented and submitted as task2.py. In order to run this, we must run in the command line: `uvicorn task2:app --reload`

We can create the following digit images using an online paint program.



The code:

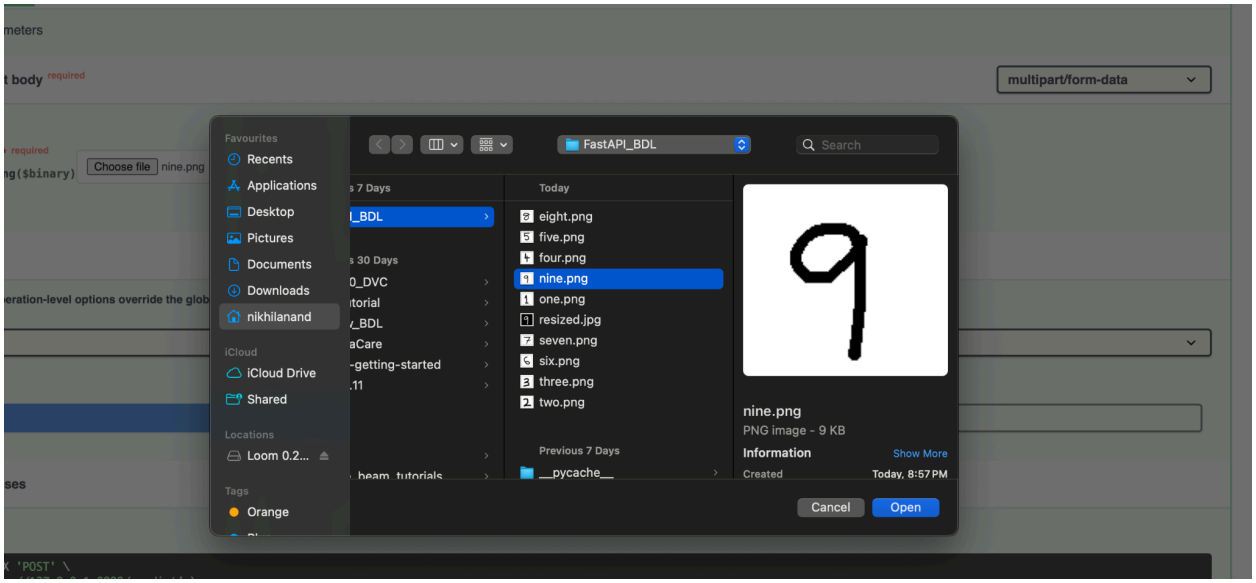
```python
@app.post("/predict")
async def predict(file: UploadFile):
    request_object_content = await file.read()
    img = Image.open(io.BytesIO(request_object_content))

    resized_img = await format_image(img)

    arr = np.array(resized_img)

    print(arr,arr.shape)

    flattened_image=arr.reshape(-1)
    flattened_image_list = flattened_image.tolist()

    model = await load_model("/Users/nikhilanand/FastAPI_BDL/training_1/cp.weights.h5")
    digit = await predict_digit(model,flattened_image_list)

    return {"digit":digit}
```

First we import the image through the post http function. Then the image is formatted. Then we flatten it and load the model and predict the digit. The digit is shown in the Swagger UI as a dictionary {“digit”: digit”}

You can see the remaining functions in the code file which is submitted along with this report.

Results

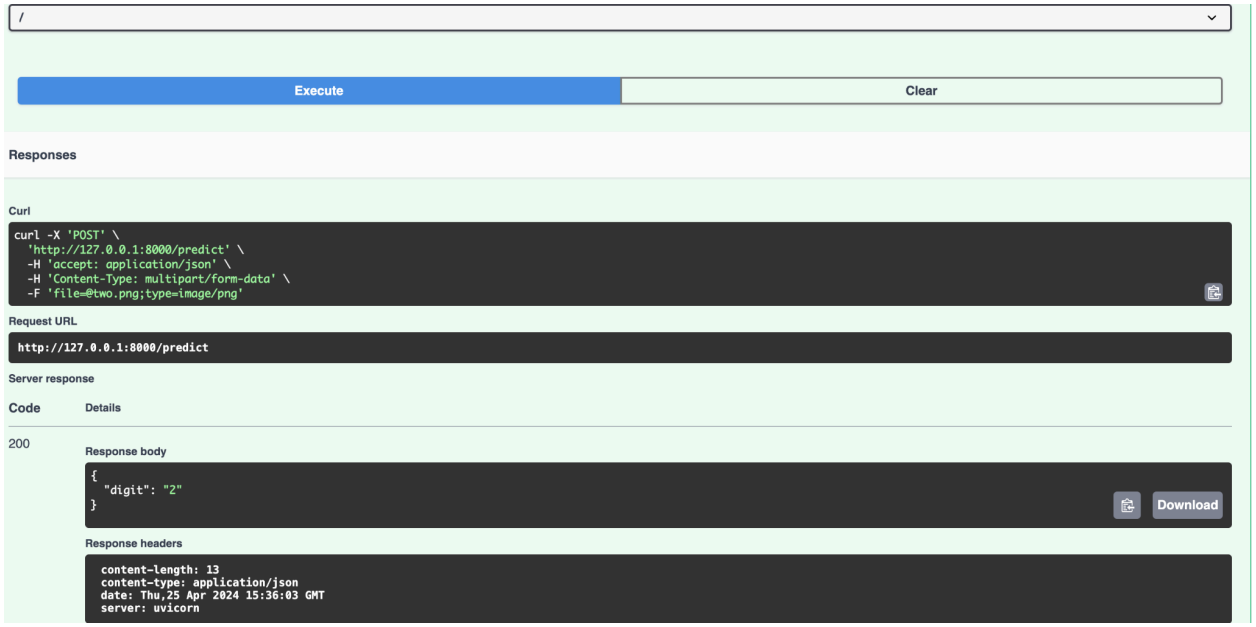
Loading the 9:



It outputs 9:

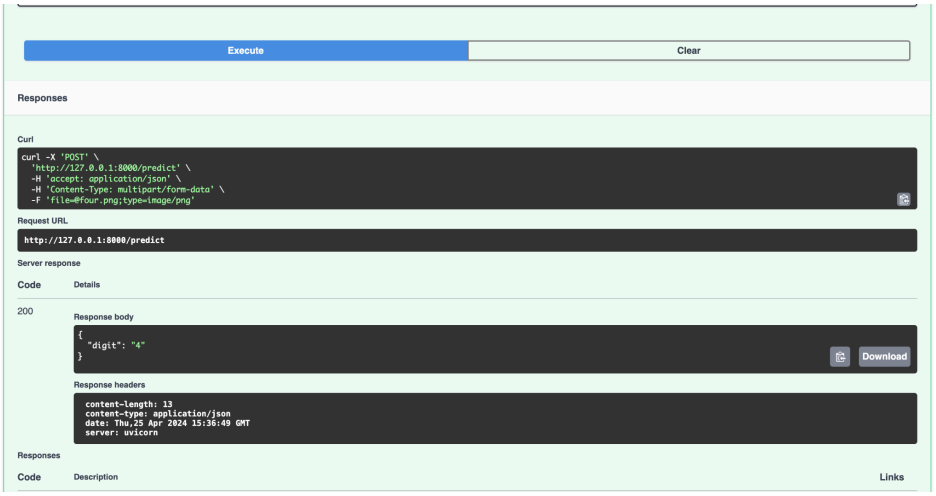
Loading the 5,

Loading the 2,



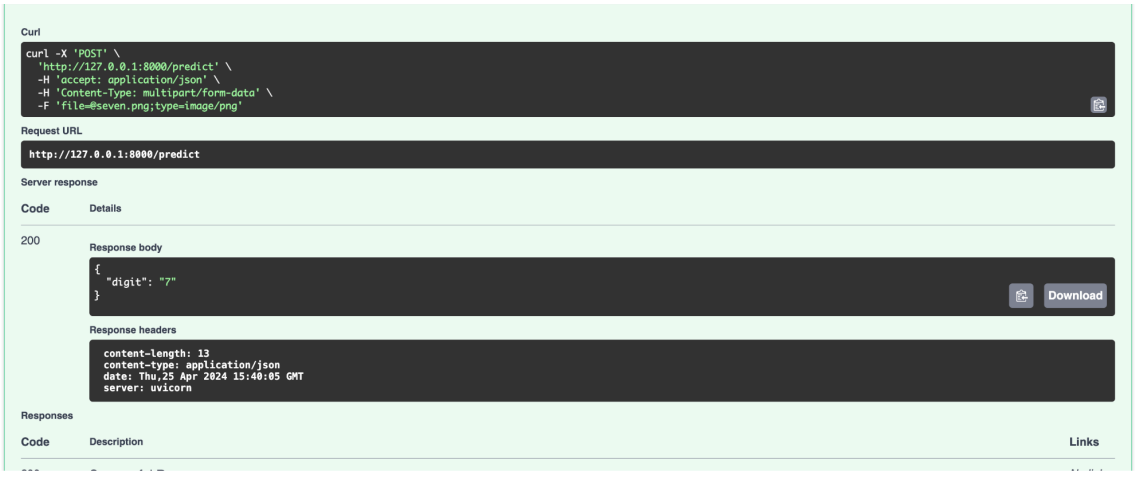
Loading the 3,

Loading the 4,



Loading the 8,

Loading the 7,



Loading the 1,

Thus, **our model worked for all 10 digits!**
