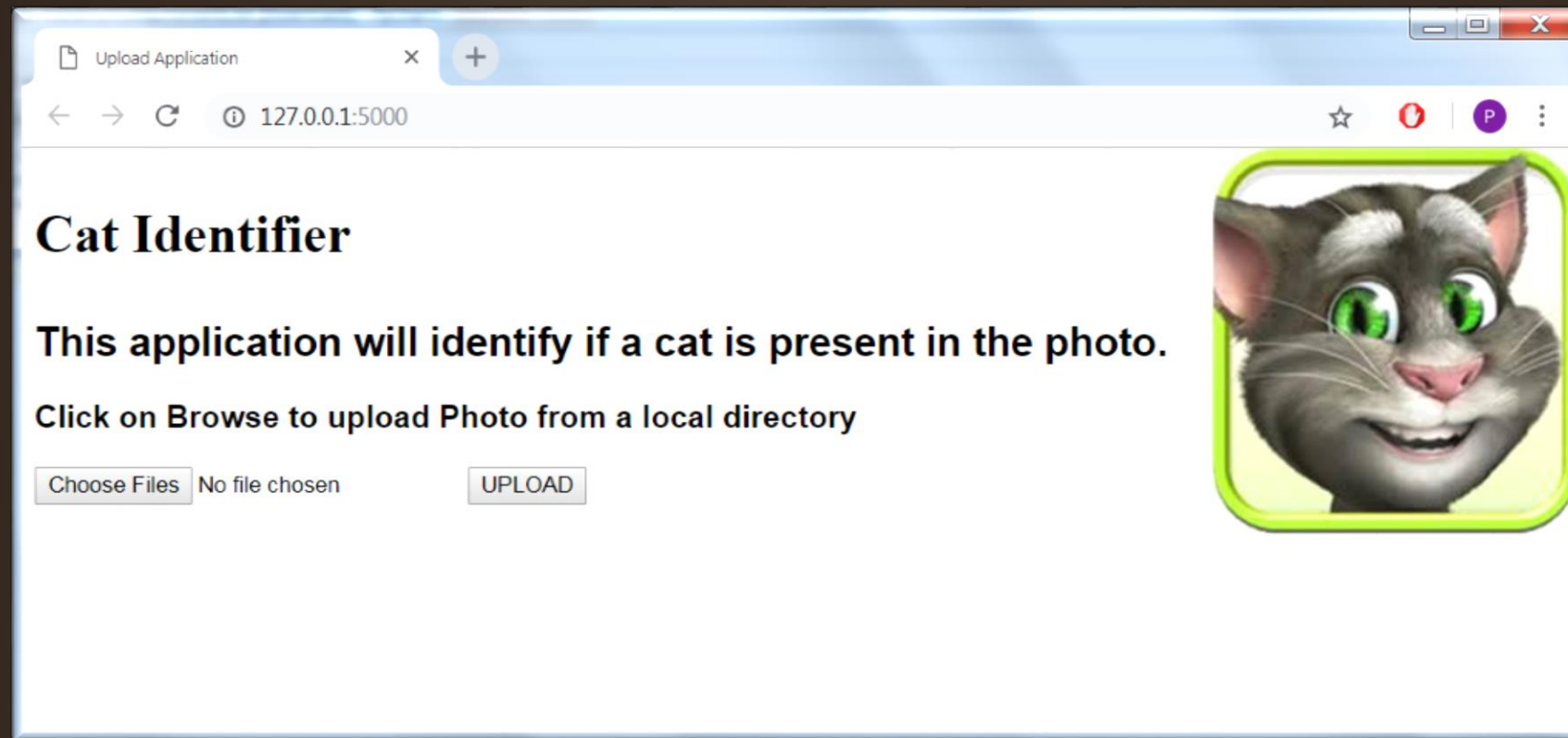


Cat Detector Application



Presenter: Prerna Prateek

Split the problem into three parts

- ▶ Application for Image Upload and Save to disk
 - Designed Front-End using HTML and CSS
 - Used “form” tag and constrained on only File of type Image to get uploaded
- ▶ Machine Learning Model to identify if Image has Cat
 - Used Google Cloud Vision API
- ▶ Method to send saved Image to Machine Learning Model
 - Backend was created in Flask in Python 3

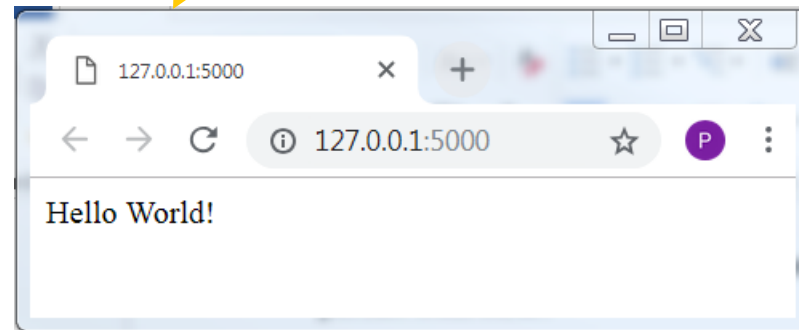
Google's Cloud Vision API

- ▶ Easy to create API key and using its RESTful API
- ▶ Has high Rate Limiting (free 1000 images/day)
- ▶ Built-in Pre-trained Models for Object Detection
- ▶ Generates labels detected within the Image
- ▶ Associates a Score with each label
- ▶ Gives Scope to alter and change the Model for Object Detection

How Flask creates App!

- ▶ `from flask import Flask`
- ▶ `app = Flask(__name__)`
- ▶ `@app.route("/")`
- ▶ `def hello():`
 - ▶ `return "Hello World!"`
- ▶ `if __name__ == "__main__":`
 - ▶ `app.run()`

```
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Key Components of Flask Used

- ▶ **render_template()**: When provided with the name of the template and the variables as keyword arguments, it renders a template. Flask will look for templates in the templates folder
- ▶ **request()**: Helps application to access the file from the files dictionary on the request object
- ▶ **url_for()**: Builds a URL to a specific function

Description of Built-in Python Library

- ▶ **os:** module provides a portable way of using operating system dependent functionality
 - mkdir
 - path
 - join/isdir/abspath/dirname
- ▶ **base64:** provides functions for encoding binary data to printable ASCII characters and decoding such encodings back to binary data
- ▶ **google-api-python-client:** Enables accessing Google API
- ▶ **GoogleCredentials:** Takes JSON file of credentials and authorizes to use a web service

Request-Response Cycle

- ▶ Flask App Sends Requests to Cloud Vision API
 - Request has encoded image data and information about Model to use on image
- ▶ Cloud Vision API returns Response with processed information
- ▶ Python code handles the Response and gets prediction
- ▶ Returns to Flask App to display Result in Front-End

Importance of Pytest

- ▶ Helps to document expected behavior of Code
- ▶ Fixtures can be created once and used multiple times
- ▶ Ensures one part of Code Change does not break/affect other parts
- ▶ Gives facility to “mock” certain parts of code for creating unit-tests

Thank You!