Flask + Bootstrap

- I used flask and bootstrap to built a 2 page web application which collects the information about donors and also display the same in another page.
- I haven't used any database to store or extract information so all the data you are seeing on the website is dummy data.
- My intent is to show that flask can render HTML pages based on the routes given to the application.

Blog on Flask

- Flask is a Python-based micro web framework. It is characterized as a micro framework since it does not necessitate the usage of any special tools or libraries.
- It lacks a database abstraction layer, form validation, or any other components that rely on pre-existing third-party libraries to perform common operations.
- Flask, on the other hand, enables extensions that can add application functionalities as if they were built into Flask itself.
- Extensions are available for object-relational mappers, form validation, upload handling, several open authentication systems, and a number of framework-related utilities.
- Flask’s framework is more explicit than Django and is also easier to learn because it has less base code to implement a simple web-Application.
- Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine. Below is the basic starter code of a flask app.

```python
# an object of WSGI application
from flask import Flask
app = Flask(__name__)   # Flask constructor

# A decorator used to tell the application
# which URL is associated function
@app.route('/')
def hello():
    return 'HELLO'

if __name__=='__main__':
   app.run(debug = True)
```

- ‘/’ URL is bound with `hello()` function. When the home page of the web server is opened in the browser, the output of this function will be rendered accordingly.
- The Flask application is started by calling the `run()` function. The method should be restarted manually for any change in the code. To overcome this, the debug support is enabled so as to track any error.
- One can also use flask as a back end processing python program (functions) for many Machine Learning Algorithms. And the same usage can be extended for making database transactions such as inserts , updates, deletes etc.
- Best Minimal Resource: https://flask-restful.readthedocs.io/en/latest/quickstart.html
- Best Maximal Resource: https://realpython.com/tutorials/flask/ (paid tho)

Blog on Bootstrap:

- Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
- Bootstrap's CSS can be accessed by the HTML elements by using the exact names as class names of tags.
- These class names are available in the documentation of bootstrap. (https://getbootstrap.com/docs/4.6/getting-started/introduction/). Look at the below snippet.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content=
        "width=device-width, initial-scale=1">

    <link rel="stylesheet" type="text/css"
        href="css/bootstrap.min.css">

    <script src="js/bootstrap.min.js"></script>
</head>

<body>
    <div class="container text-center">
        <h1 class="text-success">Data Foundation Systems</h1>
        <p>A course for a task</p>
    </div>
</body>

</html>
```

- As you can see there is a file called bootstrap.min.css which can be downloaded from the website. You can also use inbuilt web links for using the CSS required.
- The class names "container text-center" are the standard names which will be mentioned in the documentation.
- There are many other plugins and extensions of bootstrap which can be used with front-end JavaScript frameworks. And this saves a lot of time for web developers to manage their development schedule.
- Best Minimal Resource: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp
- Best Maximal Resource:https://getbootstrap.com/docs/4.6/getting-started/introduction/

Blog on Keras

I have created a simple RNN and LSTM model for creating a model to predict the sentiment of a movie. I used IMDB data set here.

- Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation.
- Keras is **Simple** -- but not simplistic. Keras reduces developer *cognitive load* to free you to focus on the parts of the problem that really matter. For example there is no need to implement a backtracking algorithm which was already implemented and that too with different variations based on the inputs given by us.
- It is **Flexible** -- Keras adopts the principle of progressive disclosure of complexity: simple work flows should be quick and easy, while arbitrarily advanced work flows should be *possible* via a clear path that builds upon what you've already learned.
- It is **Powerful** -- Keras provides industry-strength performance and scalability: it is used by organizations and companies including NASA, YouTube, or Waymo.
- The core data structures of Keras are **layers and models**. The simplest type of model is the Sequential model, a linear stack of layers.
- For more complex architectures, you should use the Keras functional API, which allows to build arbitrary graphs of layers, or write models entirely from scratch via sub classsing.

This is a basic sequential model

```
from tensorflow.keras.models import Sequential

model = Sequential()
```

We can add layers to the model with a simple add() function.

```
from tensorflow.keras.layers import Dense

model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
```

Once the required design is implemented we can start compiling the model we have designed until now. later we can used the fit() function to start feeding the training data to the model. And then we can predict the test data using the predict function.

```
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.SGD(learning_rate=0.01, momentum=0.9, nesterov=True))
model.fit(x_train, y_train, epochs=5, batch_size=32)
classes = model.predict(x_test, batch_size=128)
```

- Best Minimal Resource: https://www.tensorflow.org/resources/learn-ml
- Best Maximal Resource: https://keras.io/getting_started/intro_to_keras_for_engineers/

REACT JS + MONGO-DB + EXPRESS

- In this application I mimicked what I have done in the initial Flask+Bootstrap one.
- What is extra here is that , there is actually a database which is storing the information in the backend
- To prove the efficiency of react , I implemented a simple sort function, which sorts the data in the frontend.
- To run this code you might need react JS and a MONGO DB community server installed on your local machine.
- The backend is basically a express server which maintains a constant connection with the backend mongoDB server to handle the database operations which are sent by API calls via react frontend.
- The frontend basically collects information from the admin and stores the data in the database, which can be retrieved. More about MongoDB queries are mentioned in the MongoDB blog.
- Do remember to populate the data before using the data section.

Installing Node:

For Linux

```
curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -

sudo apt-get install -y nodejs
```

For Mac

```
brew install node
```

Installing MongoDB:

- Install the community edition here.

Installing React

```
npm install -g create-react-app
```

Running the code

- Run Express backend:

  ```
  cd backend/
  npm install
  npm start
  ```

- Run React Frontend:

  ```
  cd frontend
  npm install/
  npm start
  ```

- Navigate to http://localhost:3000/ in your browser.

Blog on React JS:

- React is a javascript library for building user interfaces. React makes it painless to create interactive UIs.
- Using react you can Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. This concept of Declarative views make your code more predictable and easier to debug.
- React is Component based , so you can build encapsulated components that manage their own state and not only that you can also compose them to make complex UIs.
- Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.
- React can also render on the server using Node and power mobile apps using React Native.
- React is a library for building composable user interfaces. It encourages the creation of reusable UI components which present data that changes over time.
- React approaches building user interfaces differently by breaking them into **components**. This means React uses a real, full featured programming language to render views, which we see as an advantage over templates for a few reasons:

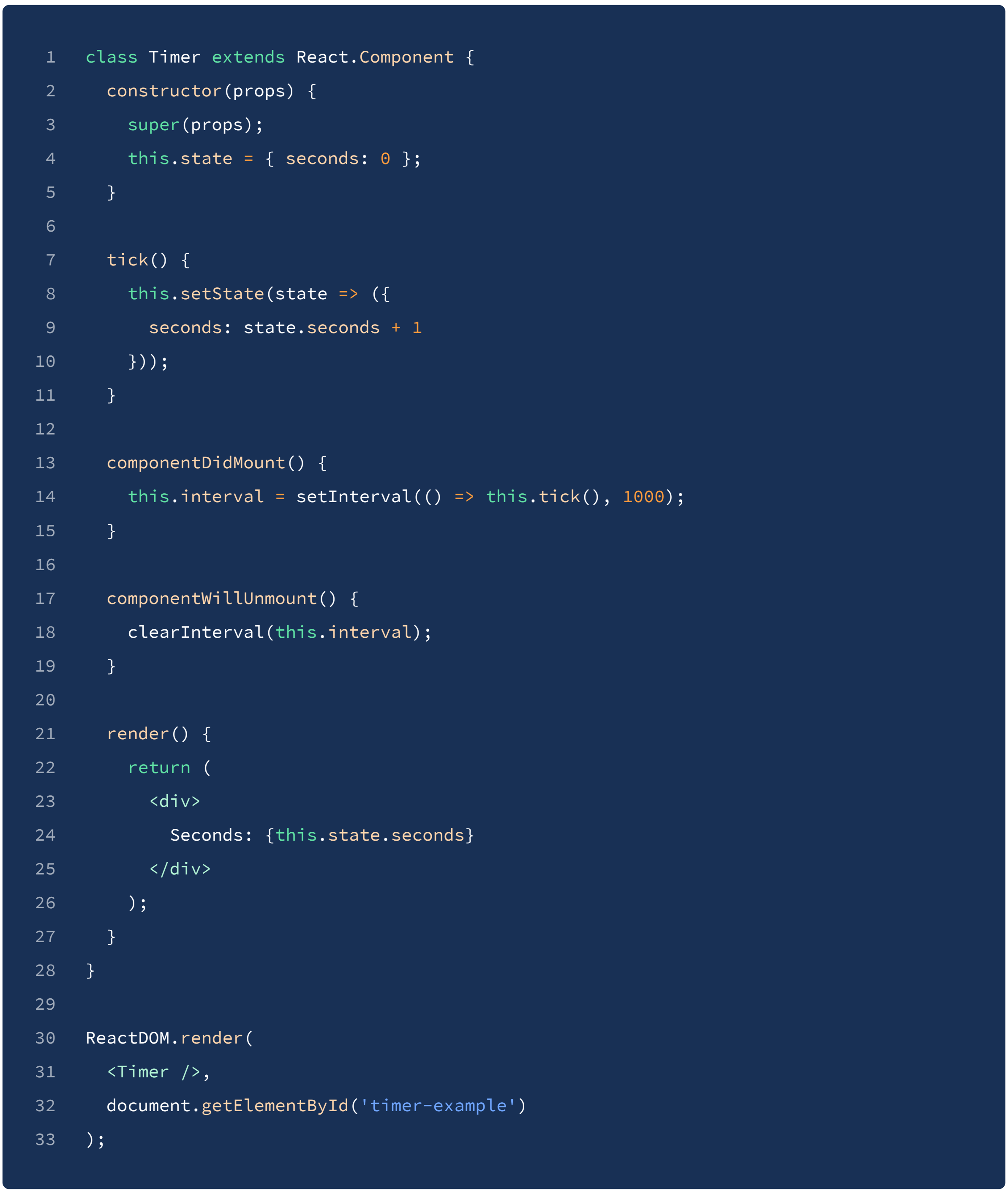
React components implement a render() method that takes input data and returns what to display. This example uses an XML-like syntax called JSX. Input data that is passed into the component can be accessed by render() via this.props.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

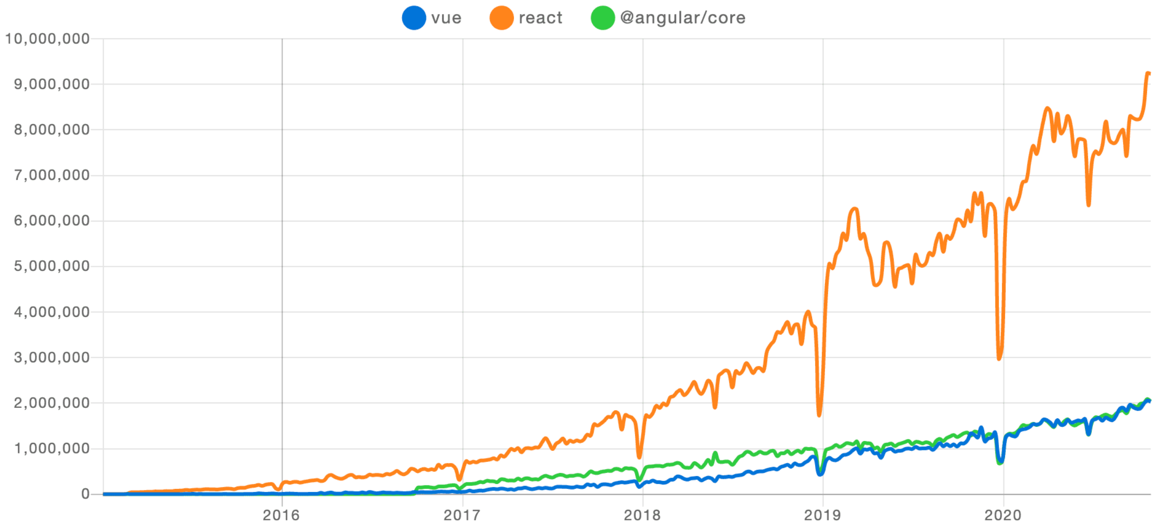
In addition to taking input data (accessed via this.props), a component can maintain internal state data (accessed via this.state). When a component’s state data changes, the rendered markup will be updated by re-invoking render().

Every time render() is called react will create a new virtual DOM where the root node is the component whose render function is called. The render() function is called when either the state **or** the props of a component or any of its children change. The render() function destroys all of the old virtual DOM nodes starting from the root and creates a brand new virtual DOM.

In order to make sure the re-rendering of components is smooth and efficient React uses the Diffing Algorithm to reduce the time it takes to create a new tree to a time complexity of $O(n)$, usually time complexity for copying trees is $> O(n^2)$. The way it accomplishes this is by using the "key" attribute on each of the elements in the DOM. React knows that instead of creating each element from scratch it can check the "key" attribute on each node in the DOM. This is why you get a warning if you don't set the "key" attribute of each element, React uses the keys to vastly increase its rendering speed.



```
class Timer extends React.Component {
  constructor(props) {
    super(props);
    this.state = { seconds: 0 };
  }

  tick() {
    this.setState(state => ({
      seconds: state.seconds + 1
    }));
  }

  componentDidMount() {
    this.interval = setInterval(() => this.tick(), 1000);
  }

  componentWillUnmount() {
    clearInterval(this.interval);
  }

  render() {
    return (
      <div>
        Seconds: {this.state.seconds}
      </div>
    );
  }
}

ReactDOM.render(
  <Timer />,
  document.getElementById('timer-example')
);
```



- Best Minimal Resource: https://www.tutorialspoint.com/reactjs/reactjs_quick_guide.htm
- Best Maximal Resource: https://reactjs.org/docs/getting-started.html

Blog on MONGO-DB

- MongoDB is a document-oriented NoSQL database used for high volume data storage.
- Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents.
- Documents consist of key-value pairs which are the basic unit of data in MongoDB.
- Collections contain sets of documents and function which is the equivalent of relational database tables.
- MongoDB is a database which came into light around the mid-2000s.
- Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other.
- The document structure is more in line with how developers construct their classes and objects in their respective programming languages.
- The rows (or documents as called in MongoDB) doesn’t need to have a schema defined beforehand. Instead, the fields can be created on the fly.
- The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.
- MongoDB offers a community server for mac, windows and Linux. Apart from this MongoDB also offers cloud solutions where our allocated servers can be accessed by a security key. (More about this https://www.mongodb.com/atlas/database)
- Scalability – The MongoDB environments are very scalable. Companies across the world have defined clusters with some of them running 100+ nodes with around millions of documents within the database
- You can go through this below code snippet which does some basic operations by querying the Mongo server installed on our PC.

```
// This command displays which database we are using currently
> db
test

// We are forcing our system to use a different database named tutorial
> use tutorial
switched to db tutorial

> db
tutorial

// This command displays the documents stored in a collection named donations
> db.donations.find()
{ "_id" : ObjectId("61f28e7302ba729336686ce9"), "name" : "swamy Naidu", "email" : "swamymagnet@gmail.com", "amount" : "1000", "__v" : 0 }
{ "_id" : ObjectId("61f30aabe3edc0fa5bc6a67f"), "name" : "Vinay Thumpala", "email" : "swamynaidu.ch@research.iiit.ac.in", "amount" : "8000", "__v" : 0 }
{ "_id" : ObjectId("61f31180e3edc0fa5bc6a681"), "name" : "Yuva Mani Teja", "email" : "yuva.mani@students.iiit.ac.in", "amount" : "10000", "__v" : 0 }


// We are deleting one single record from the database where the amount is 10000
> db.donations.deleteOne( { amount: "10000" } )
{ "acknowledged" : true, "deletedCount" : 1 }



> db.donations.find()
{ "_id" : ObjectId("61f28e7302ba729336686ce9"), "name" : "swamy Naidu", "email" : "swamymagnet@gmail.com", "amount" : "1000", "__v" : 0 }
{ "_id" : ObjectId("61f30aabe3edc0fa5bc6a67f"), "name" : "Vinay Thumpala", "email" : "swamynaidu.ch@research.iiit.ac.in", "amount" : "8000", "__v" : 0 }

```

```


// We are inserting a new record here.
> db.donations.insertOne({"name" : "Sreehith", "email" : "sreehith.manubolu@gmail.com", "amount" : "5000", "__v" : 0 })
{
    "acknowledged" : true,
    "insertedId" : ObjectId("61f313bb733ffe276d072413")
}



// This is just a command for displaying the output beautifully
> db.donations.find().pretty()
{
    "_id" : ObjectId("61f28e7302ba729336686ce9"),
    "name" : "swamy Naidu",
    "email" : "swamymagnet@gmail.com",
    "amount" : "1000",
    "__v" : 0
}
{
    "_id" : ObjectId("61f30aabe3edc0fa5bc6a67f"),
    "name" : "Vinay Thumpala",
    "email" : "swamynaidu.ch@research.iiit.ac.in",
    "amount" : "8000",
    "__v" : 0
}

{
    "_id" : ObjectId("61f313bb733ffe276d072413"),
    "name" : "Sreehith",
    "email" : "sreehith.manubolu@gmail.com",
    "amount" : "5000",
    "__v" : 0
}

> db.donations.insertOne({ "name" : "Yuva Mani Teja", "email" : "yuva.mani@students.iiit.ac.in", "amount" : "10000", "__v" : 0 })
{
    "acknowledged" : true,
    "insertedId" : ObjectId("61f31180e3edc0fa5bc6a681")
}


// If you can see there is a error in email of Vinay Thumpala, So we are updating this here.
> db.donations.updateOne( {name: "Vinay Thumpala"},{$set:{email:"vinay.thumpala@research.iiit.ac.in"} })
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }



//Now you can see the updated email of Mr.Vinay Thumpala here.
> db.donations.find().pretty()
{
    "_id" : ObjectId("61f28e7302ba729336686ce9"),
    "name" : "swamy Naidu",
    "email" : "swamymagnet@gmail.com",
    "amount" : "1000",
    "__v" : 0
}
```

```
{
    "_id" : ObjectId("61f30aabe3edc0fa5bc6a67f"),
    "name" : "Vinay Thumpala",
    "email" : "vinay.thumpala@research.iiit.ac.in",
    "amount" : "8000",
    "__v" : 0
}
{
    "_id" : ObjectId("61f31180e3edc0fa5bc6a681"),
    "name" : "Yuva Mani Teja",
    "email" : "yuva.mani@students.iiit.ac.in",
    "amount" : "10000",
    "__v" : 0
}
{
    "_id" : ObjectId("61f313bb733ffe276d072413"),
    "name" : "Sreehith",
    "email" : "sreehith.manubolu@gmail.com",
    "amount" : "5000",
    "__v" : 0
}
```

NOTE: I feel we should refer the documentation of every technology before referencing any other 3rd party resources and I can justify my statement with the level of familiarity one can get with the docs of some technologies like React JS, Redux, Mongo DB etc.

Just making a point here because it is my own Blog!!!!!!!!!!!!

React vs Flutter

React can be applied to android and IOS apps developments in the form of react native. Gradually the number of third party libraries we can use along with react are increasing and making the lives of developers easy. But React is not alone. There is also another technology called FLUTTER which is gaining massive popularity in the recent times due to the same reason. We can say an average undergrad developer should know at least one of the above two frameworks. You may already be aware that React Native uses JavaScript. This is a well-known programming language that most developers are already used to. In fact, According to Stack Overflow, over 70% of pro developers are familiar with JavaScript. When you move into using React, it is intuitive if you've previously used JavaScript.

Flutter uses the programming language Dart, which is almost exclusive to Google. It does allow you to compile native code at a more rapid rate, but it means a steeper learning curve for anyone outside of Google.

In this case, most people prefer to stick with what they already know, so React Native tends to be chosen.

	Flutter	React Native
What is it?	A portable UI toolkit for building natively-compiled apps across mobile, web, and desktop* from a single codebase	A framework for building native applications using React
Official release	December 2018, Google I/O	March 2015, F8 Conference
Created by	Google	Facebook
Programming language	Dart	JavaScript
Popularity	120,000 Stars on Git hub (May 2021)	95,300 stars on Git hub (May 2021)
Hot Reload	Yes	Yes
UI	**Flutter apps look as good on the up-to-date operating systems as they do on older versions.**Since they only have one codebase, the apps look and behave similarly across iOS and Android – but thanks to Material Design and Cupertino widgets, they can also imitate the platform design itself. How's that possible?Flutter contains two sets of widgets which conform to specific design languages: Material Design widgets implement Google's design language of the same name; Cupertino widgets imitate Apple's iOS design.**This means that your Flutter app will look and behave naturally on each platform, imitating their native components.**	**Application components look just like native ones** (e.g. a button on an iOS device looks just like a native iOS button, and the same on Android).The fact React Native uses native components under the hood should give you confidence that, after any OS UI update, your app's components will be instantly upgraded as well.That said, this can break the app's UI but it happens very rarely.**If you want your app to look near-identical across platforms** – as well as on older versions of an operating system (as Flutter achieves) – **then consider using third-party libraries** (like this one). They will enable you to use Material Design components, in place of native ones.
Sharing code	With Flutter 2 (announced in March 2021), we can use the same codebase to ship native apps to five operating systems: iOS, Android, Windows, macOS, and Linux; as well as web experiences targeting browsers such as Firefox, Chrome, Safari, or Edge. Flutter can even be embedded in cars, TVs, and smart home appliances. (*source*).Perhaps the single largest announcement in Flutter 2 is production-quality support for the web. It can be used for: **Progressive Web Apps (PWAs)** that combine the web's reach with the capabilities of a desktop app,**Single Page Apps (SPAs)** that load once and transmit data to and from INTERNET services. **Existing mobile apps** – allowing Flutter apps to be run on a desktop.Find out more here.	**iOS and Android** – but there are select libraries that allow you to use the same code to build **iOS, Android, web, and Windows apps**.You can also extract shared code in mobile, desktop, and web apps, to a separate repository; treat it as a separate project; then inject it in the same way as another dependency. This allows a developer to focus on writing code for a specific platform without having to consider compatibility with another one.

SQL vs MongoDB

It is a well known fact that SOL databases have ruled the world of data technologies and have been the primary source of data storage for over 4 decades. Generally the SOL databases are used, mainly for accessing relational databases.

Oracle and **Microsoft SQL Server** ruled the segment, but as the Web development market paced up, there came a shift towards usage of open source databases like **MySQL**, **Postgres** etc. But RDBMS was still the first choice.

Soon enough data started growing exponentially and scalability became a major issue, at that time **NoSQL** rolled in to save the day.

One of the cool facts about NoSQL databases is that they existed since 1960, but in the recent times they have gained more popularity especially for the scaling feature provided by the same.

The major difference between MongoDB and SQL Databases is the way they handle data. In SQL databases, data is stored in form of traditional 2 dimensional row-column structure while in MongoDB rich data document model is followed, which allows storage of any type of data.

Let us see some of the key differences between MongoDB and other SQL databases :

SQL Database	NoSQL Database (MongoDB)
Relational database	Non-relational database
Supports SQL query language	Supports JSON query language
Table based	Collection based and key-value pair
Row based	Document based
Column based	Field based
Support foreign key	No support for foreign key
Support for triggers	No Support for triggers
Contains schema which is predefined	Contains dynamic schema
Not fit for hierarchical data storage	Best fit for hierarchical data storage
Vertically scalable - increasing RAM	Horizontally scalable - add more servers
Emphasizes on ACID properties (Atomicity, Consistency, Isolation and Durability)	Emphasizes on CAP theorem (Consistency, Availability and Partition tolerance)

Flask vs Django

Django is a web development framework for Python. This framework offers a standard method for fast and effective website development. It helps you in building and maintaining quality web applications. It enables you to make the development process smooth and time-saving.

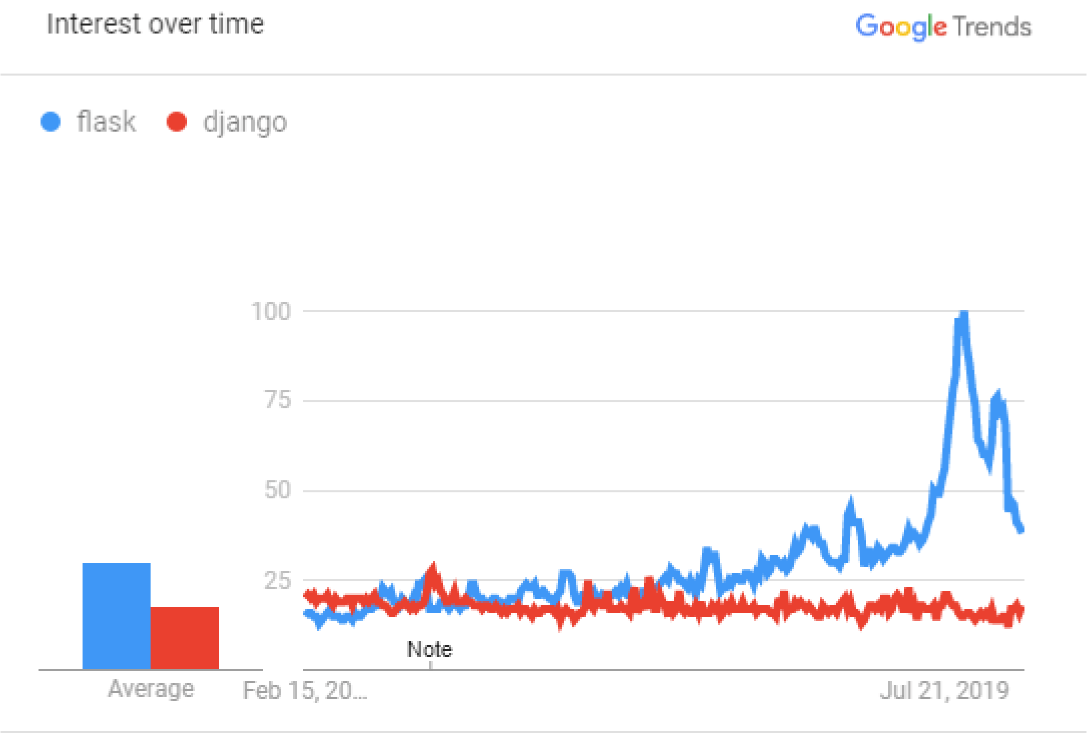
It is a high-level web framework which allows performing rapid development. The primary goal of this web framework is to create complex database-driven websites.

Flask may be a generally youthful framework, as it were in utilize since 2010. Flask is considered more “Pythonic” than Django is basically since Flask web application code is, in most cases, more unequivocal. Flask is the choice of most tenderfoots due to the need of barricades to getting a basic app up and running.

The useful admin interface is what makes Django a capable web system. Not at all like Flask, Django incorporates a ready-to-use admin system that empowers clients to carry out the extend organization errands consistently. Based on the venture, it naturally creates admin modules. Engineers can indeed customize the admin interface in arrange to meet the particular trade needs.

Not at all like Django, Flask doesn’t have a built-in layout motor. Flask is based on the Jinja2 format motor. Jinja2 is itself impacted by the Django format motor. Its employments coordinate a sandboxed execution environment, permitting engineers to speed up the advancement handle for energetic web applications. While Django incorporates a built-in format motor that permits engineers to make user-facing layers for web applications consistently and quickly.

Django has bolster for the ORM framework. Advantage of ORM framework includes: Developers can take advantage of the ORM framework to work with an assortment of databases, including PostgreSQL, SQLite, Prophet, MySQL and more. Developers don’t have to type in long SQL inquiries to execute common database operations. Whereas, Flask doesn’t supports the ORM framework. Designers are required to type in SQLAlchemy (Protest Social Mapper and SQL toolkit for Python) in arrange to perform common database operations.



Flask	Django
Created in 2010	Created in 2005
Python web framework built for rapid development.	Python web framework built for easy and simple projects.
Flask is WSGI framework.	Django is a Full Stack Web Framework.
Flask provides support for API.	Django doesn’t have any support for API.
Support Visual Debug.	No support for Visual Debug.
Flask allows you to use multiple types of databases.	Django doesn’t offer multiple types of databases.
Flask has no default support for forms, but you can use WTForms to fill the gap.	Django comes Form with which can be integrate with the ORM and the admin site.
Flask does not offer dynamic HTML pages.	Django offers dynamic HTML pages.
The request based object is imported from the flask module, which is a global variable in Flask.	All views are set as an individual parameter in the Django.
Flask is much younger platform compared to Django.	Django is a very mature framework.
Flask offers a diversified working style.	Django offers a Monolithic working style.
It supports an extension which could be implemented in the framework.	Django has its own module library. So, it stores several prewritten codes.
The structure of the project layout for Flask web framework is random.	The structure of the project layout for the Django is conventional.
Flask web framework uses a Ninja2 template design.	Django web framework helps you to utilizes the View web templating system.
URL dispatcher of the Flask web framework is a RESTful request.	URL dispatcher of this Django framework is based on controller-regex.
Flask does not offer a built-in bootstrapping tool.	Django-admin enables developers to start building web applications without any external input.
Flask is a good choice if you want a lightweight codebase.	The best feature of Django is Robust documentation.
Flask framework is suitable for single application.	Django framework allows developers to divide a project into multiple page application.
Flask Web Framework doesn’t offer support for third-party applications.	Django Web Framework supports a large number of third-party applications.
Git hub stars 48.8 K	Git hub stars 47.1 K
Best features of the flask is it is lightweight, open source, and offer minimal coding for developing an application.	The best features of Django are Rapid development, Open source, Great Community, Easy to learn.

References:

1. https://www.studytonight.com/mongodb/mongodb-vs-rdbms
2. https://www.guru99.com/flask-vs-django.html#:~:text=KEY%20DIFFERENCES%3A&text=Flask%20is%20a%20Python%20web,offers%20a%20Monolithic%20working%20style.&text=Flask%20is%20WSGI%20framework%20while%20Django%20is%20a%20Full%20Stack%20Web%20Framework.
3. https://www.thedroidsonroids.com/blog/flutter-vs-react-native-what-to-choose-in-2021
4. https://docs.mongodb.com/v4.4/mongo/
5. https://www.mongodb.com/
6. https://reactjs.org/docs/getting-started.html
7. https://keras.io/