

INFO3180 – LECTURE 8

MODEL VIEW CONTROLLER (MVC)

What is MVC?

Model View Controller (MVC) is a software architecture pattern, commonly used to implement user interfaces: it is therefore a popular choice for architecting web apps.

Source: [Mozilla Developer Network – MVC Architecture](#)

In general, it separates out the application logic into three separate parts, promoting modularity and ease of collaboration and reuse. It also makes applications more flexible and welcoming to iterations.

Source: [Mozilla Developer Network – MVC Architecture](#)

*The 3 parts are: The **Model**, the **View**
and the **Controller**.*

*The **Model** defines what data the app should contain and in many applications will represent the database.*

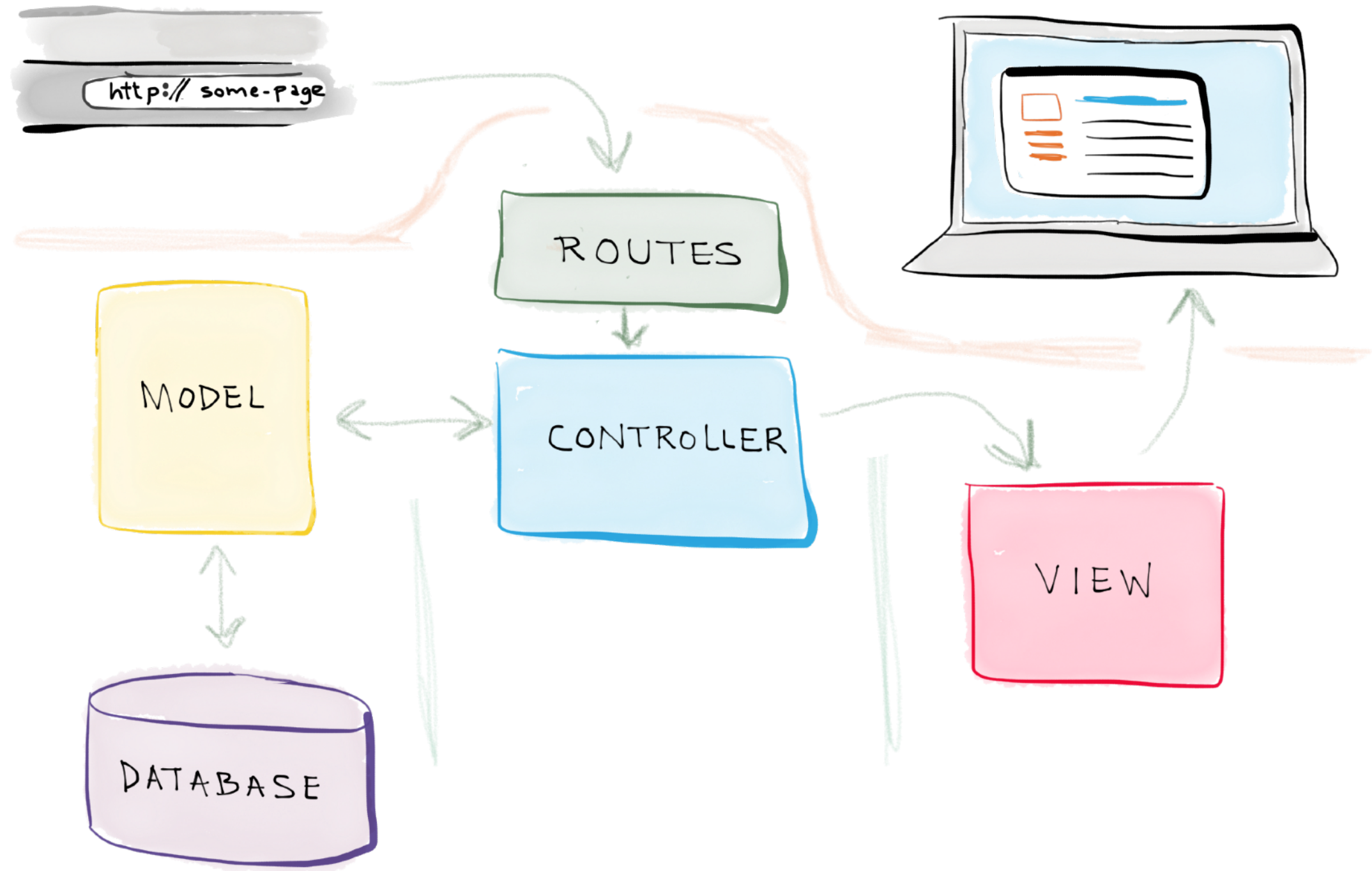
Source: [Mozilla Developer Network – MVC Architecture](#)

*The **View** is how the application's data will be displayed. In a web application this is usually the HTML templates (with help from CSS to style it).*

Source: [Mozilla Developer Network – MVC Architecture](#)

*The **Controller** contains the logic that updates the model and/or view in response to input from the users of the app.*

Source: [Mozilla Developer Network – MVC Architecture](#)



Source: [Real Python MVC Paradigm Summarized](#)

So in the context of Flask, what do the Model, the View and the Controller represent?

OUR MODEL

```
class User(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    username = db.Column(db.String(80), unique=True)  
    email = db.Column(db.String(120), unique=True)  
  
    def __init__(self, username, email):  
        self.username = username  
        self.email = email  
  
    def __repr__(self):  
        return '<User %r>' % self.username
```

Our models are the classes we define in our `models.py` file. This of course maps to a table in your database.

OUR VIEW

```
<html>
  <head>
    <title>{% block title %}{% endblock %}</title>
  </head>
  <body>
    <ul>
      {% for user in users %}
        <li><a href="{{ user.url }}">{{ user.username }}
</a></li>
      {% endfor %}
    </ul>
  </body>
</html>
```

In Flask our 'view' is our Jinja2 template files

OUR CONTROLLER

```
@app.route("/users")
def users():
    # query the model/database for users
    users = User.query.all()

    # then take the data and render in the view/template
    return render_template('users.html', users=users)
```

In Flask our 'controller' is the various routes and corresponding view functions in our `views.py` file

RESOURCES

- ▶ MVC Architecture - https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture
- ▶ Model-View-Controller (MVC) Explained -- With Legos - <https://realpython.com/blog/python/the-model-view-controller-mvc-paradigm-summarized-with-legos/>