Danai Adkission

Guest Lecture: Project and Apps

Duke CoLab
App Development

• You're in a database class. You're learning different techniques and theories on how to best manage data.
• How do you express that data to an end user in a meaningful ways?
• Web applications are a great way to express data and publish it on the web. It allows us control over how/what data is expressed, and to whom, without requiring users to understand how to use a database.
• The database is a key component of "the stack".
The Stack

• Client Side
  • Browser
  • HTML
  • CSS
  • JS

• Server Side
  • Web and/or App server (apache, nginx, gunicorn, passenger, etc.)
  • Language (Python, Ruby, NodeJS, etc.)
  • Framework (Flask, Django, Rails, etc.)
  • Database (MySQL, MariaDB, MongoDB, etc.)
Deployment

• “Turn Key” options (e.g. Heroku)
  • Usually you will need to configure your app in a specific way.
  • You do not always have full control on the host env
  • BUT “less” for you to manage

• Standard VM (e.g. Duke VCM, Digital Ocean)
  • You do all of the work
  • However because of this, you have full control

• Docker
  • Few to zero issues going to production
  • But the learning curve is REAL
  • Still need a VM, but it only needs Docker installed
Model, View, Controller (MVC)

• **View**
  • What the user sees and interacts with
  • Typically a web page

• **Controller**
  • Defines what logic should run given the path and request
  • In most cases will call on logic that lives in a model
  • Provides results and directs traffic to new view

• **Model**
  • Typically defines database objects and performs CRUD actions
  • Also where most of your business logic lives
Model, View, Controller (MVC)
Flask

- Barebones web development framework (and I mean BAREbones)
- "Full Stack"

```python
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```
Innovation Co-Lab

- Resource on campus for technology and engineering projects.
- Part of the Office of Information Technology (OIT)
- Workshops and consultation on ongoing projects for students, staff, and faculty
Co-Lab Resources for 316

• Co-Lab office hours [https://colab.duke.edu/resources](https://colab.duke.edu/resources)
  • Student staffed hours
  • FTEs available by appointment

• Roots Courses
  • 1-3 hour workshops on various topics
    • Web frameworks
    • Programming languages
    • Design
    • Etc

• Advising with Danai (da129@duke.edu)
Helpful Links

• Flask documentation: http://flask.palletsprojects.com/en/1.1.x/
• Co-Lab Website: https://colab.duke.edu
• SQLAlchemy: https://docs.sqlalchemy.org/en/13/
• Bootstrap: https://getbootstrap.com/
• jQuery: https://api.jquery.com/
• Example flask app: https://gitlab.oit.duke.edu/colab/flask_template
• https://reactjs.org/
• https://vuejs.org/