A Case Study of the Python Django Framework to Help Recruiters Identify Analytics Talent

Xin Li Leong (leong8@purdue.edu), Matthew A. Lanham (lanhamm@purdue.edu)

Purdue University, Krannert School of Management

Abstract

We develop a web application for companies seeking analytics talent for Purdue's M.S. in Business Analytics & Information Management program using the python Django framework. The motivation for this work is that university career services functions are often poor at showcasing their students for employment. Secondly, many data science and analytics degree programs either lack a course or opportunities within the curriculum that provide students an opportunity to design and develop open-source front-end analytics tools for their end users. Our study provides a case for such a course where a student might extend their python programming skills to develop a front-end GUI using the Django framework for a real critical need within a universities' career services function.

Introduction

Often there is a disconnect between the university program and career services. University programs spend a great deal of their marketing resources highlighting faculty, school branding, showcasing innovative curriculum, or other initiatives that help pull students in the door. The problem is once students are ready to enter the job market, they are supported by a career services function that rarely helps recruiters or hiring managers find what they are looking for. The services provided mostly focus on resume and cover letter reviews and mock interviews. The issue is there is a missing step in the placement process where those hiring need to be able to seek and narrow down what they are looking for among candidates that are still seeking employment. Having the ability to do this better would help close the loop from students entering the program to graduating from the program. We posit this could also better help achieve academic program placement statistics.
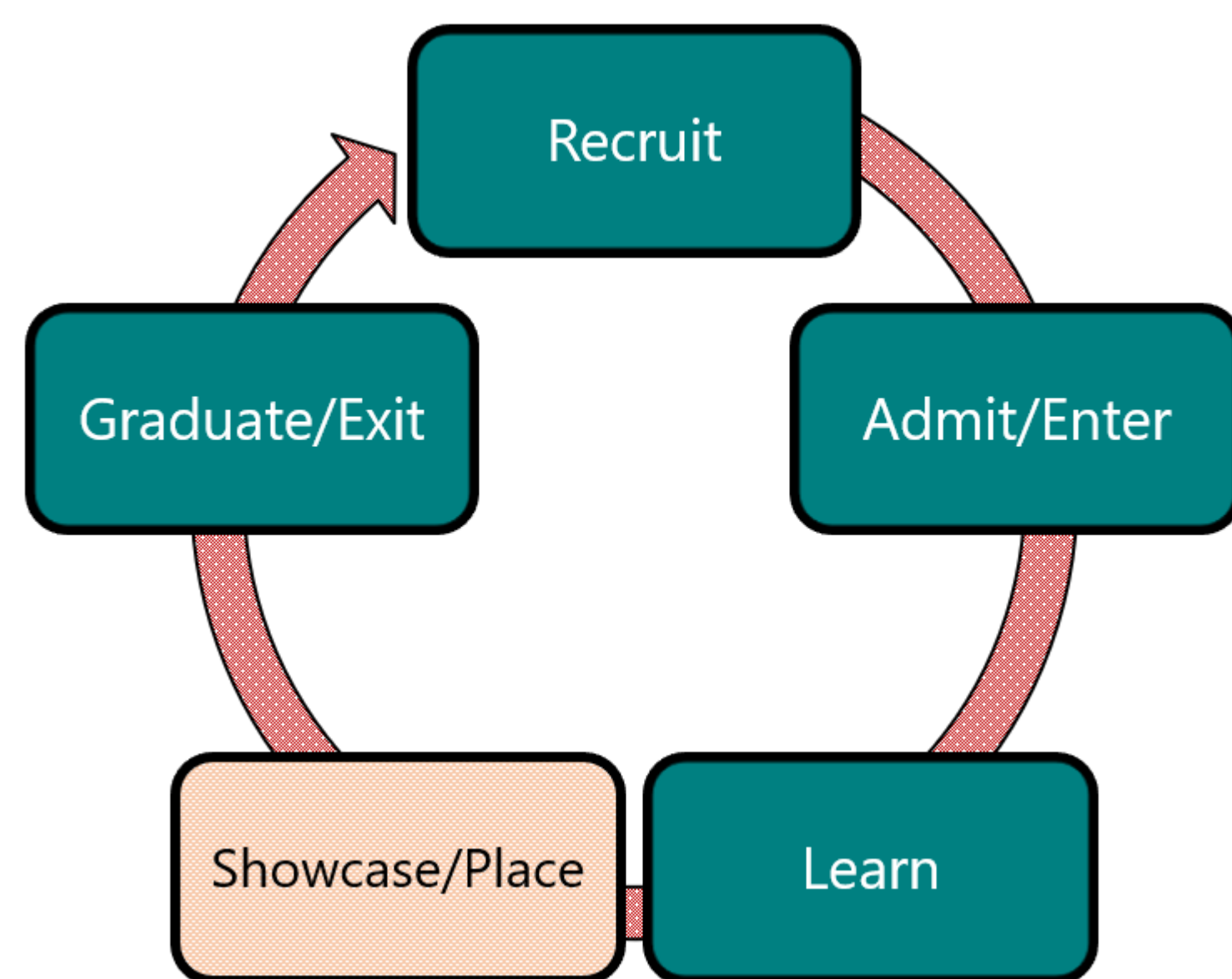
Many data science and analytics programs teach students how to develop a data product or tool in R using Shiny, but very few, show best practices or examples of how to develop front end tools using Python.

Research Question

Can we provide a case example of the Django framework for future analytics students that helps support a career services and academic program need for current analytics students?

Data

Data used in this study was provide by candidates in Purdue's M.S. in Business Analytics & Information Management program who were set to enter the job market in May 2020.

Website Design Considerations

User Interface Design — We drafted a website layout that's user friendly, incorporated Purdue's signature colors and was unique compared to other websites to attract recruiter's attention. Recruiters could filter passed on certain candidate features.

Data Design — We drafted an ERD containing all required tables and fields taking special consideration into handling already placed students.
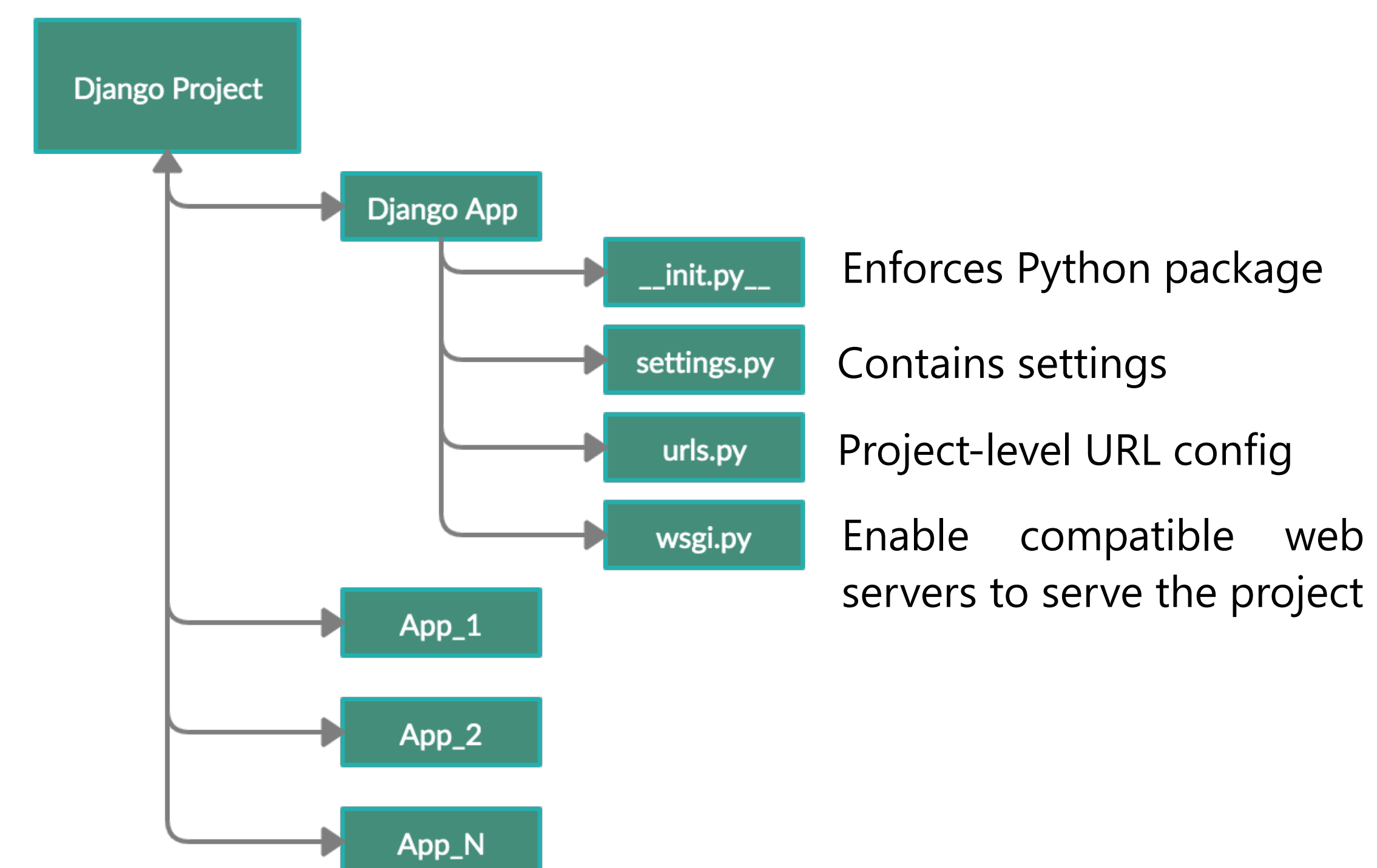
Django Framework

Django has an MVT architecture which shares similarities with the MVC architecture. The MVT architecture is unique where a View is a request handler while any other app logic that does not return a response is placed in a separate file.
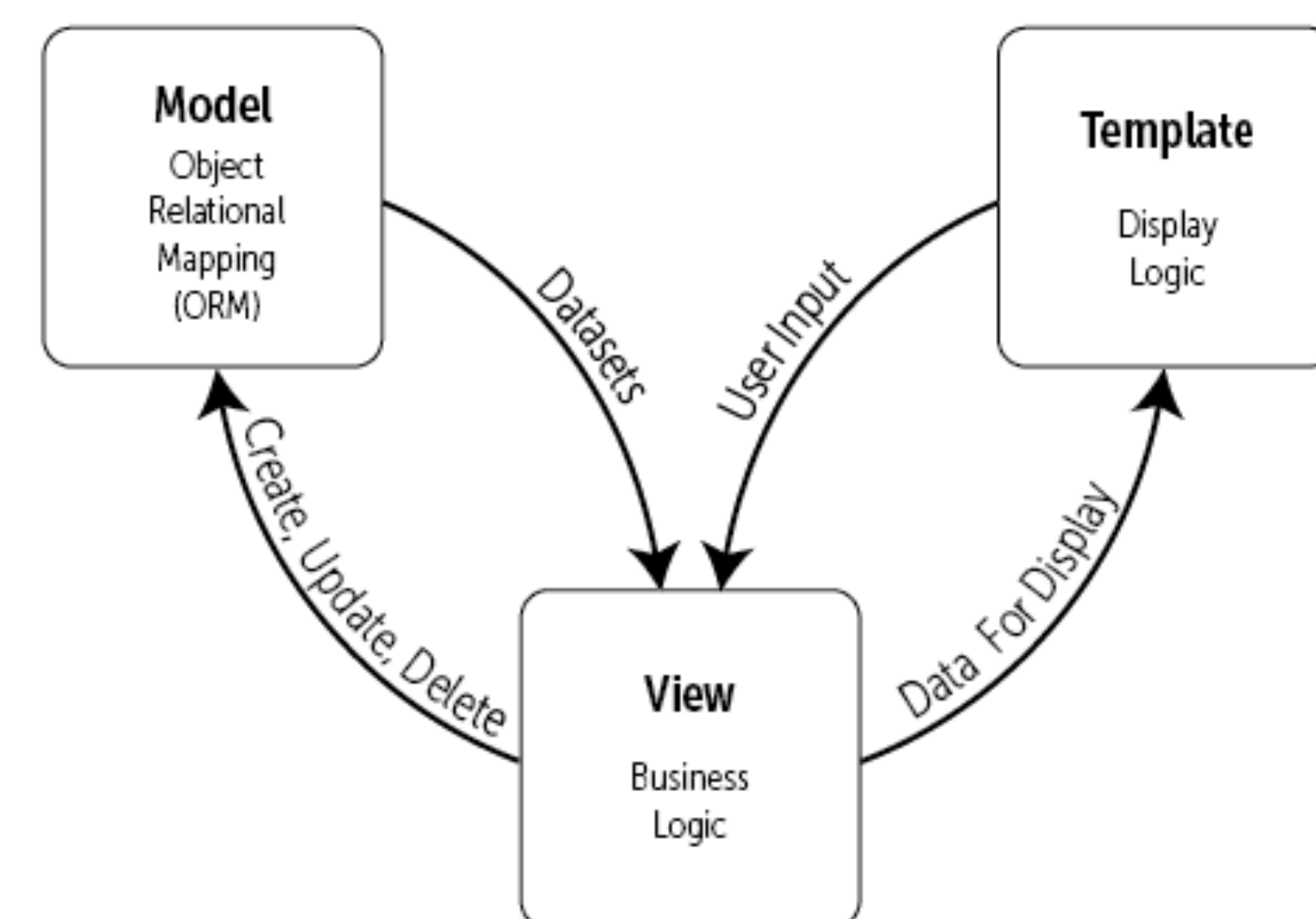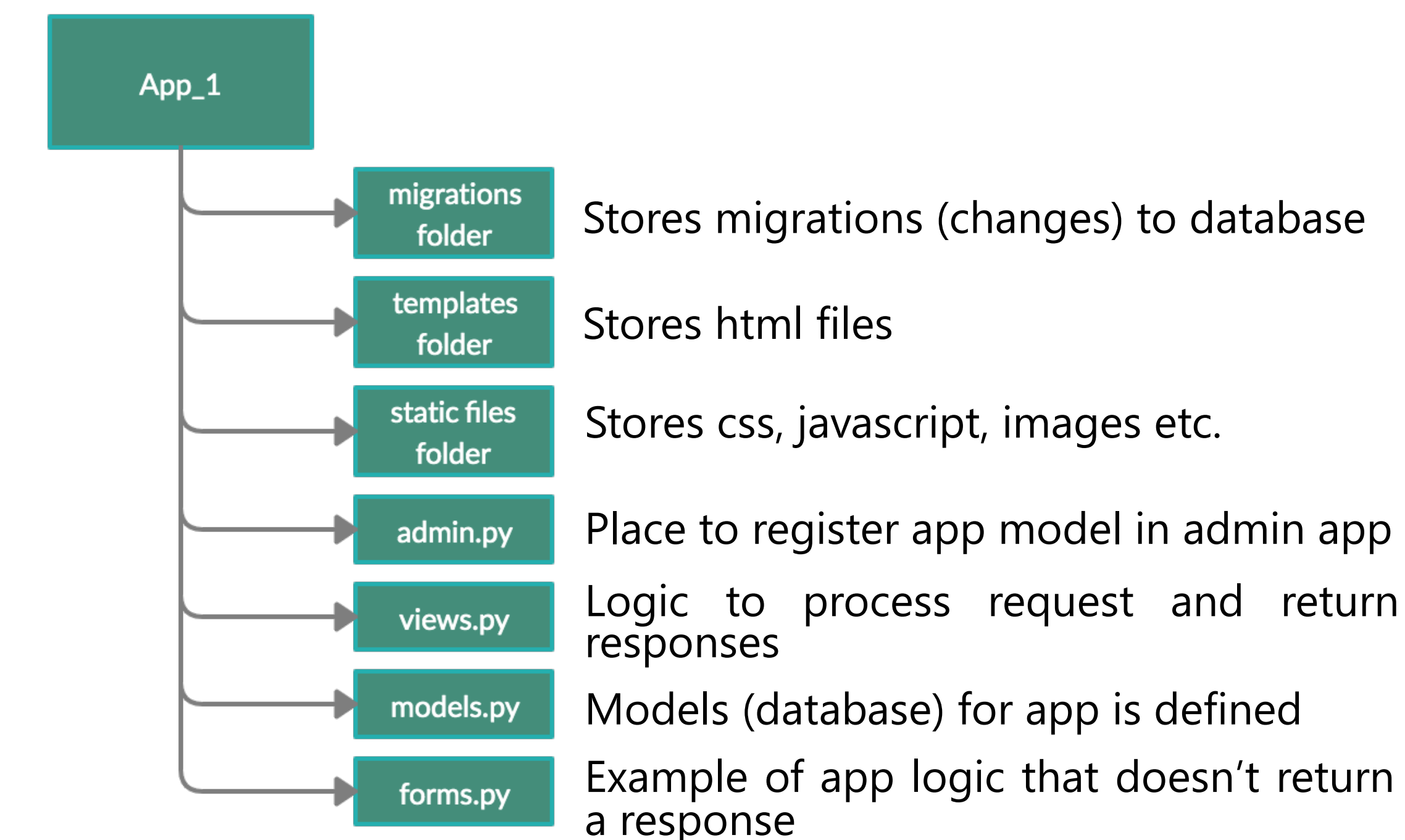
Django Project Setup

1. Install Python3
2. Set up a virtual environment
3. Install Django
4. Create your project
   - ***django-admin startproject projectname***
5. Starting an application
   - ***python manage.py startapp appname***

Django Project Structure

- Django Project
  - Django App
    - __init.py__ — Enforces Python package
    - settings.py — Contains settings
    - urls.py — Project-level URL config
    - wsgi.py — Enable compatible web servers to serve the project
  - App_1
  - App_2
  - App_N

Django Self-Created App Structure

- App_1
  - migrations folder — Stores migrations (changes) to database
  - templates folder — Stores html files
  - static files folder — Stores css, javascript, images etc.
  - admin.py — Place to register app model in admin app
  - views.py — Logic to process request and return responses
  - models.py — Models (database) for app is defined
  - forms.py — Example of app logic that doesn't return a response
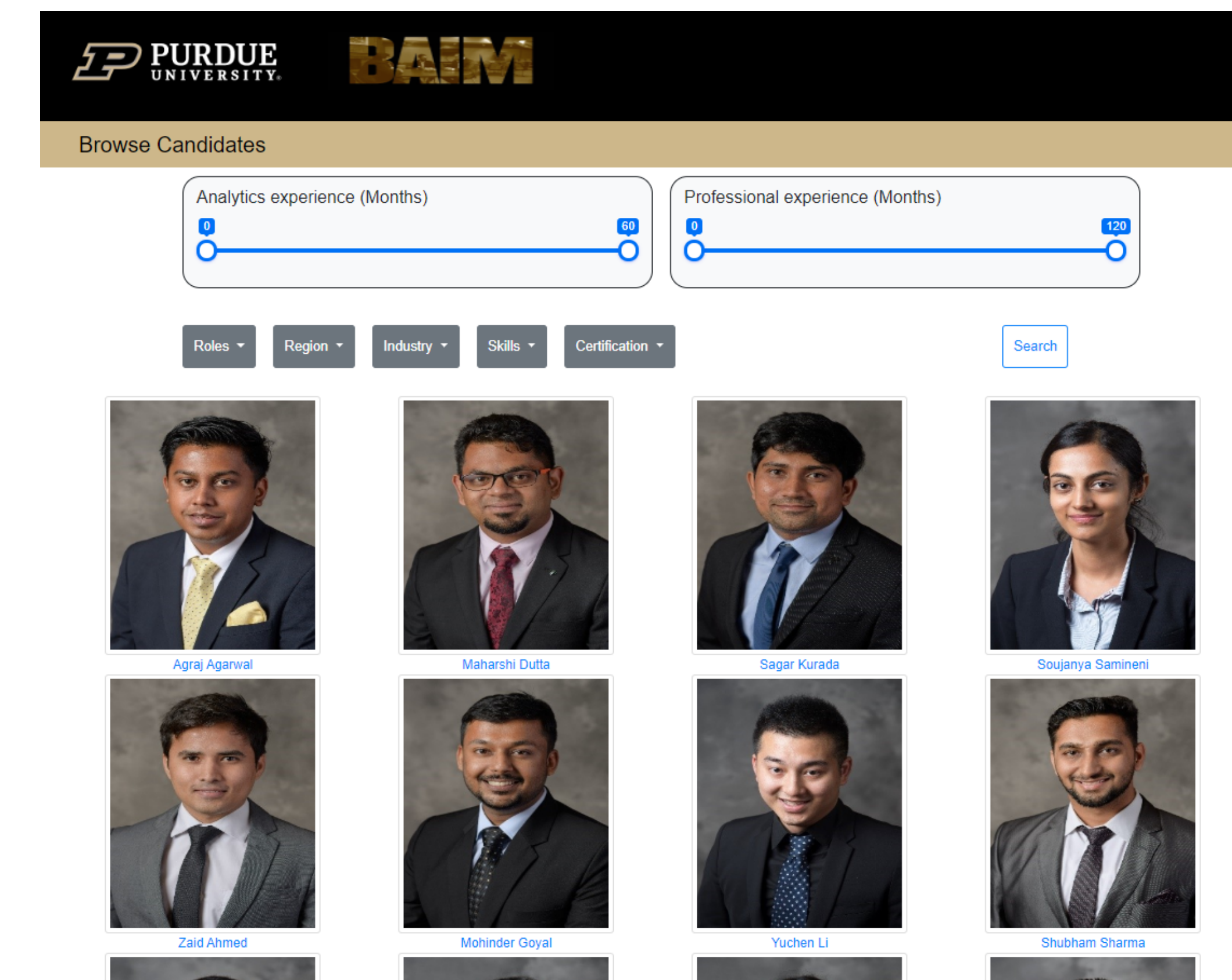
Django Benefits

1. **Complete**
   Django packages contain nearly everything developers would need to use under one framework. Everything works seamlessly together and has an extensive up-to-date documentation and a strong online community support as well.
2. **Secure**
   Django's framework has been built to protect the website automatically. It handles account and password security, CSRF security and many other things,
3. **Scalable**
   Django's independent project structure allows developers to change or replace any individual component easily without minimal effect to the rest of the project.
4. **Maintainable**
   Django code design prioritizes maintainable and reusable code and uses the DRY principle.
5. **Pre-built Apps**
   Django has a lot of useful apps such as ***django-registration***, ***django-import-export*** and ***django-crispy-forms*** that can be easily integrated into your project without much extra coding.

Working App

http://testaccountbaim.pythonanywhere.com/candidates/

GitHub Repository

https://github.com/lxlxinli/purdue-BAIM-website

Conclusions & Future Features

This study provides a working example of the python Django framework to help career services better showcase Purdue's MS BAIM students.

We plan to add the following features to this project then deploy it for use this May. We will survey industry partners and the students to obtain feedback on the applications use in helping match them to jobs.

- Remove candidates that have accepted offers
- Add placement locations and program statistics
- Provide interactive Tableau figures
- Provide a short video from each candidate about themselves like a dating service
- Show candidate images as retro baseball cards

Acknowledgements