Names: Jacob Smith, Ailish Skinner, Kody Quintana, Jacob Kohav, Noah Svensson, Milo Boyer

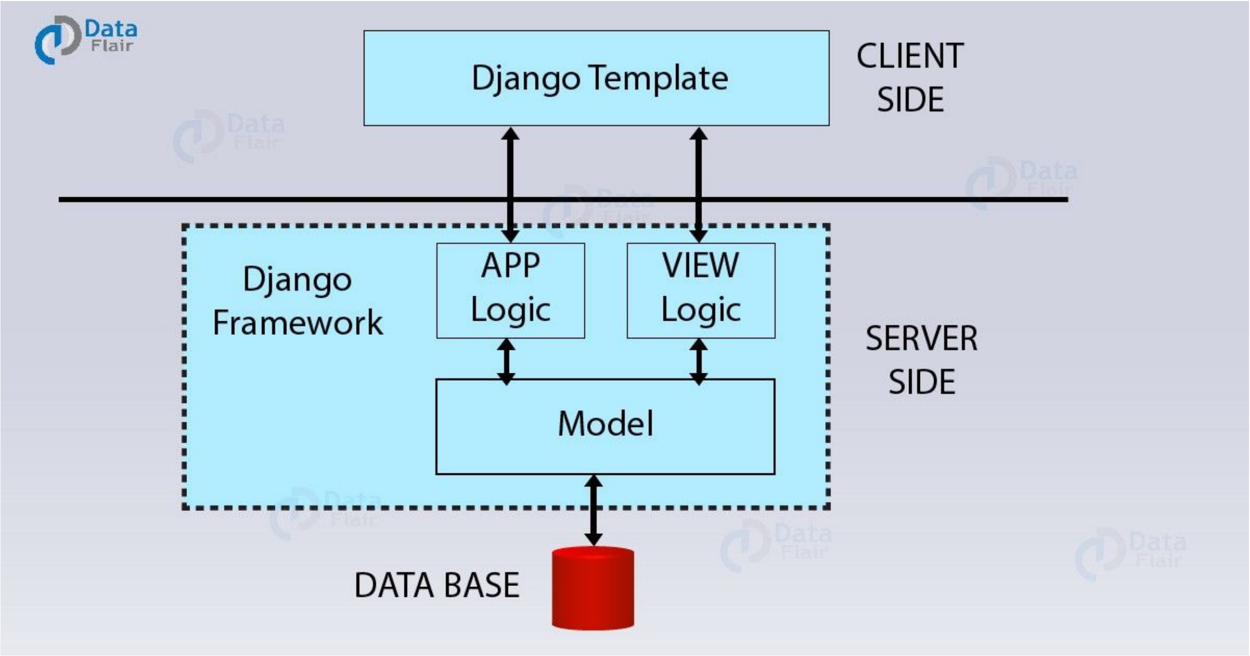
Group: 204-1 “Number 1”

Revised List of Features:

Features:

1. **Register a Company**
   A “login” page that will let you register your company and allow employees/employers to access the quiz.
2. **Register Employee**
   Similar to Register a Company, however, it will allow the employees to be able to sign up and choose the employer to store and access the employee data
3. **Slide/Embedded Video**
   Will have the tutorial sequence that you can click through until you complete it and get to the quiz. (Do you need to be registered for this?)
4. **Quiz on Content**
   A standard quiz page on the material of the slides. Current format is one question appears, then once answered the correct answer and explanation is displayed. After that, the next question is displayed on the page, and this format follows until all questions have been answered. Only accessible if you are registered
5. **Saving Place in Slideshow**
   Captures where the user was last on within the slideshow and be able to access the point they left off at.
6. **Employee Readiness Rep**
   Lists all employees (only viewed from the HR portal) that belong to an organization and then shows whether they have completed the training or not.

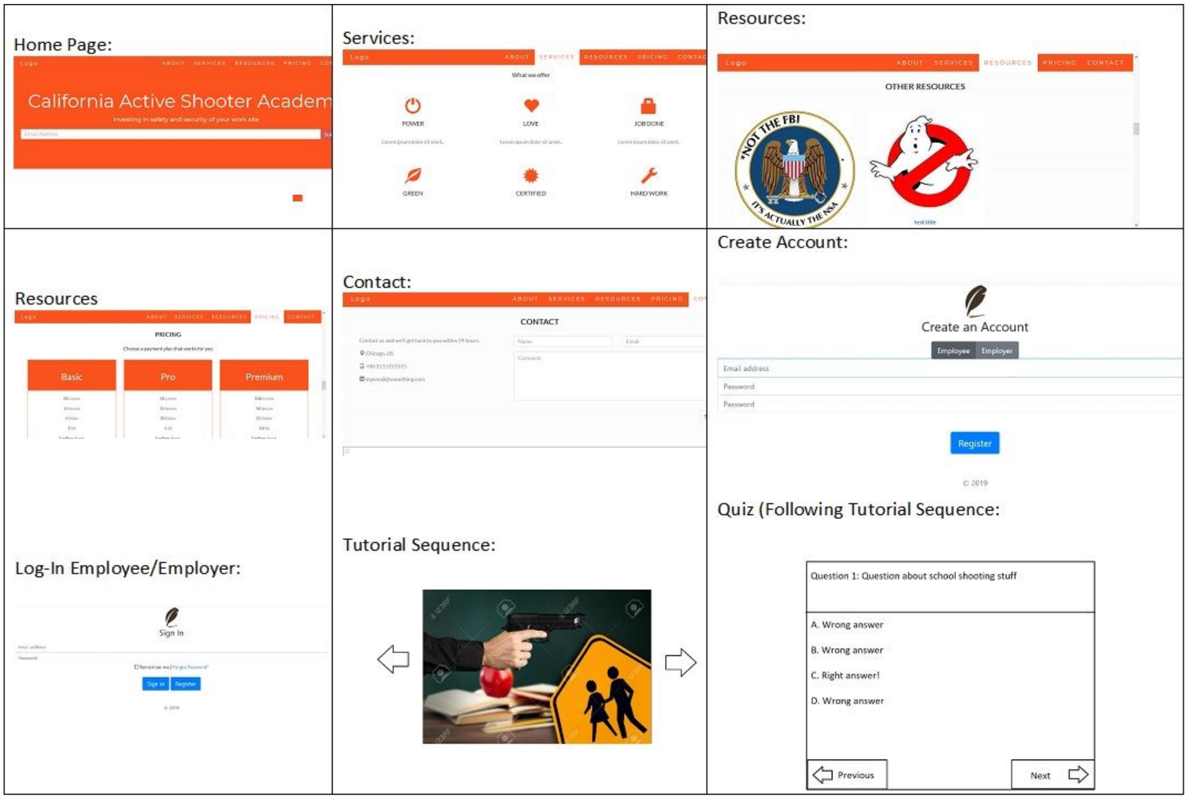
Architecture Diagram:



Django's Model is an abstraction for database tables. Currently, we are using SQLite, because it is the default database that comes with Django. We are also using Django's web server, if we we're to deploy this both the database and web server should be changed to a dedicated server and database but the Django provided ones are fine for development. Django uses a version of the model, view, control paradigm. Django's templates differ from the jade/pug templates we used in lab7 in that they are more restrictive on having logic in them. Basic for loops are allowed but things like if else statements are instead moved to the model class. For example, if you wanted to conditionally add some data to a template you would create a method in the relevant class that would be called from that template.

Front-End Design

1. Wireframe

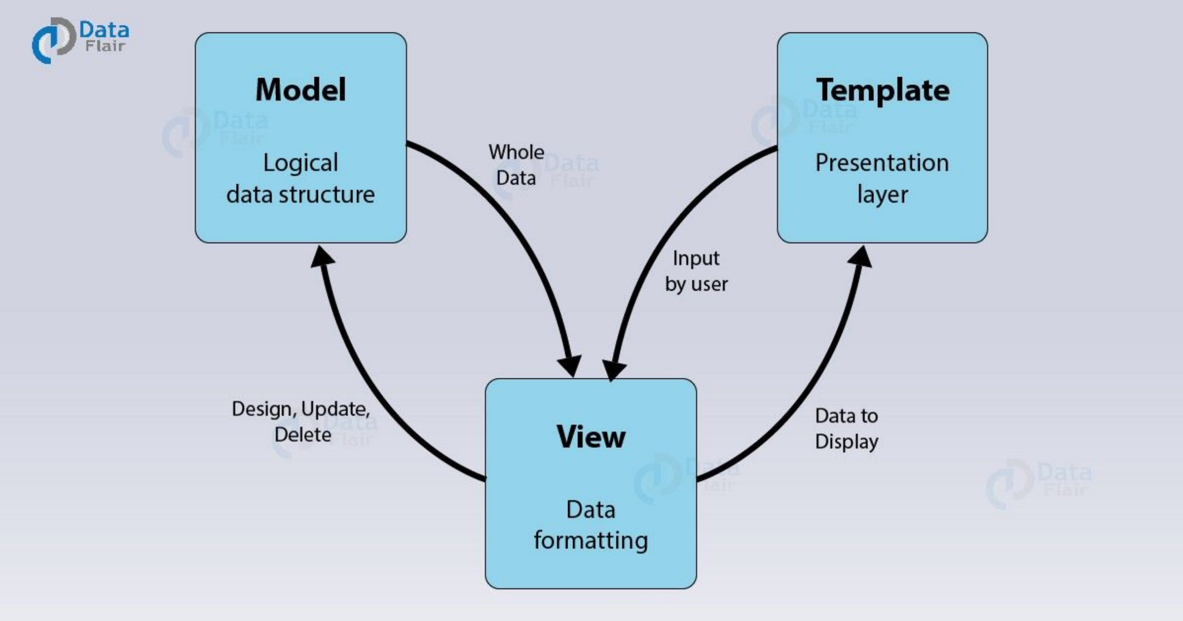


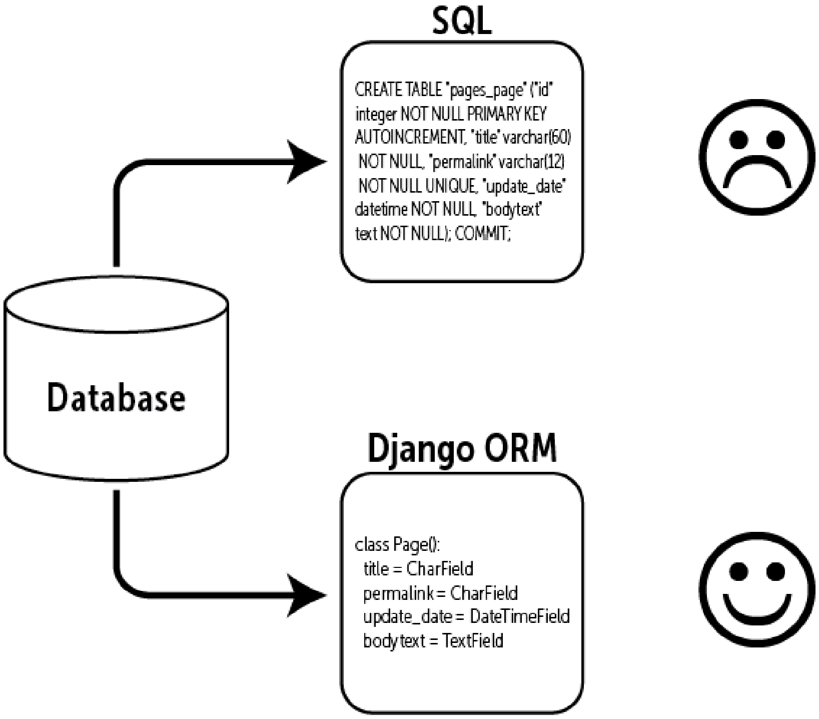
Web Service Design: N/A

Database Design

1. SQLite - predominantly read-only or require a smaller installation footprint

We are using SQLite only because it is the default database provided by Django for development purposes. A better database like postgresql can be easily configured to work with Django. A benefit of using Django is that it abstracts away interactions with the database to something more "pythonic". A database table is configured by creating a python class that inherits from Django's model base class. Instances of that class are like entries in that table. The diagram on the next page illustrates the easier interface of Django's model compared to the equivalent sql query.





```
CREATE TABLE "pages_page" ("id"
integer NOT NULL PRIMARY KEY
AUTOINCREMENT, "title" varchar(60)
 NOT NULL, "permalink" varchar(12)
 NOT NULL UNIQUE, "update_date"
datetime NOT NULL, "bodytext"
text NOT NULL); COMMIT;
```

```
class Page():
  title = CharField
  permalink = CharField
  update_date = DateTimeField
  bodytext = TextField
```