Ross Cutler

Mark Guindi

Danny Janani

Harish Saravanakumar

Juhi Singh

Navneet Singh

**Explanation of Project Files**

Please see the document on installing the software necessary for running the code.

The main folder which contains our project code is ‘evalproject’, which contains two folders ‘evalproject’ and ‘eval’, as well as a file ‘manage.py’. the manage.py file is a framework file (don’t touch it!), used to exectute various commands.

The internal folder evalproject contains the general files for any project. This contains settings.py, which contains the settings for the project (database settings are found here, as well as other settings). It also contains urls.py, which contains the general list of urls that the user navigates to, and the methods that handles how to respond to those user requests. This file will justr redirect to the other ‘urls.py’ file in the ‘eval’ folder, which is specific to the evaluation system application.The evalproject folder also contains other general files for use by the framework, that are of no concern to us.

The ‘**eval’** folder is the folder that contains the bulk of the code for the evaluation system application.

The **models.py** file contains the schema of the database, in the form of Python classes. It contains information for the Professor, SemesterCourse, Evaluation etc. tables, and the fields of those tables. The classes also contain methods to make things easier, such as the ability to represent objects as strings.

The admin.py file just lists which Models are to be managed through the admin panel.

The **urls.py** file in the ‘eval’ folder lists all the urls used by the application, which directs user actions to the appropriate front-end and back-end response. For example, when the professor clicks on a button to generate a report, the click of the button is mapped to a url, and urls.py file says which function to call that generates the information for the report, generates the webpage containing the report, and sends the report back to the user.

The **views.py** file contains the functions which handle the business logic. They create, update, and delete, rows/entries in the database, and decide which HttpResponses (webpages) to return, as well as some of the content of those webpages. Although the file is quite long, we tried splitting it up into different files with different logical parts, but things were breaking, and we couldn’t figure it out in time, so we just kept it all in one file.

The **templates** folder contains an ‘eval’ folder which contains HTML templates (with a bit of Javascript), which is basically the HTML of each webpage (these are the webpages that the views.py file renders and sends). They are templates, because they are given information, and generate unique webpages based on that information.

The migrations folder contains files that are generated automatically by the framework. If changes to the database schema are made, the corresponding SQL code is generated and run when migrations are made, and they are found in this folder.

The **static** folder just contains static files, such as CSS styles, or static images, and other assets.

tests.py is used for tests, and we did not find this file necessary.

Any other folders or files, such as \_\_init\_\_.py, wsgi.py, \_\_pycache\_\_, apps.py, etc. are for framework use, and are not used by us.

The commands to use for initial setup, and for starting the web server, are found in the “**Evaluation System Installation Instructions**” document.