

Centre for Cloud Computing and Big Data

Nikhil Ram Vikas Gowda Rakshith C Dhruva Kashyap

Cloud Based Evaluation Policies

Project Report

Introduction

We were tasked to build an evaluation system for the Computer Science Department at PES University. This evaluation system was to be used by faculty to evaluate Final Year projects. Evaluations are conducted as follows.

- Students are required to form teams and select a faculty member as a guide and register with the project co-ordinator of the Comp Sci. Department. If no guide is chosen, the project co-ordinator assigns one to them.
- Upon completion of registration of teams, the project co-ordinator will create panels and assign faculty members to each of these panels. The project co-ordinator will then assign one faculty member from each panel as the co-ordinator for a panel which we refer to as a panel-cordinator.
- The co-ordinators of each panel are in charge of scheduling reviews for all the teams in their panel. They are also responsible for assigning faculty members to review teams for each review.
- The **evaluator** performs the job of grading each team assigned to them for every review

Features of the website

Implementation

1. Back End
 - Rest Based API
 - The Back-End of the Web Application is a Rest Based API which interacts with the Front-End using JSON based requests and responses
 - Security

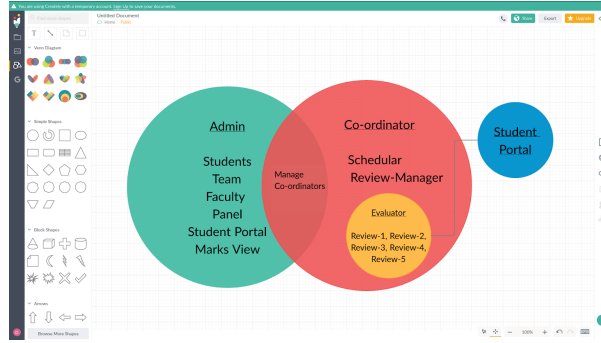


Figure 1: User Roles

- The Web Application Authenticates the user using a combination of JWT tokens and Session Cookie.
 - The JWT Access Token is used to authenticate the user to interact with the REST API and hence its validity is kept short.
 - The Access Token is refreshed periodically using a Refresh Token whose validity is substantially longer.
 - Refresh Token for the corresponding user is stored at the Back End and is accessed using the Session Cookie.
 - Roles
 - The Interface User's are
2. Single Page Application
 3. Easy Interface for Users
 4. User Authentication Using JWT Tokens
 5. Dockerized for reliability across cross platforms
 6. Rest API incorporated with PostgreSQL and Django Web Framework
 7. Security against XSS

Requirements

| Packages | Version | Details |
|-------------------------------|---------|---------------------------|
| PostgreSQL | v12.4 | Database Used With Django |
| Python | 3.6 | |
| psycopg2-binary | v2.8.5 | PostgreSQL API for python |
| Django | v3.0.8 | Python Framework |
| django-bleach | v0.6.1 | To Prevent XSS attacks |
| djangorestframework | v3.11.0 | REST API |
| djangorestframework-simplejwt | v4.4.0 | For JWT Tokens |
| Pillow | v7.2.0 | Image Processing |

| Packages | Version | Details |
|----------------|-----------|-------------------------------|
| uWSGI | v2.0.19.1 | To Deploy Django With Nginx |
| Docker | v1903.6 | |
| docker-compose | v1.17.1 | Combine Containers |
| Gunicorn | v19.7.1 | Python WSGI HTTP Server |
| Nginx | v1.14.0 | Load Balancer for HTTP server |

More detailed list is available in the source code as `requirements.txt`

Run the following commands

```
docker-compose up
```

The dockerfile is as follows

```
FROM python:3.6
EXPOSE 8000
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
ADD requirements.txt /code/
RUN pip install -r requirements.txt
ADD . /code/
LABEL creators="the NVRD team"
```

The Actual Website

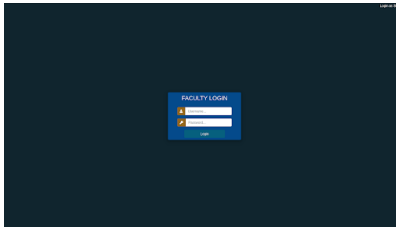


Figure 2: Faculty Login

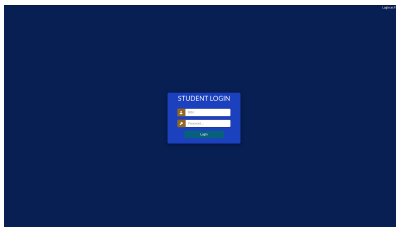


Figure 3: Student Login