

# Centre for Cloud Computing and Big Data

Cloud Based Evaluation Policies

BC\_457\_903\_957\_972

Project Report

Dhruva Kashyap

Rakshith C

Vikas Gowda

Nikhil Ram

## Abstract

Our project for the 2020-summer internship at CCBBD was to design a web application to ease the massive amount of physical paper work involved in evaluating Final-Year B.Tech Projects at the Computer Science Department at PES University.

## Table of Contents

Introduction . . . . .	1
Evaluation procedure . . . . .	1
Structure of the solution . . . . .	2
Implementation . . . . .	3
Requirements . . . . .	4
The Actual Website . . . . .	6

## Introduction

Our task was to build an evaluation system for the Computer Science Department at PES University. This evaluation system is to be used by faculties to evaluate final year projects as per the *evaluation procedure*.

### Evaluation procedure

- Students are required to form teams and select a faculty member as a guide and register with the **project coordinator** of the Computer Science Department. If no guide is chosen, the project coordinator assigns one to them.
- Upon completion of registration of teams, the project coordinator will create **panels** and assign faculty members to each of these panels. The project coordinator will then assign one faculty member from each panel as the coordinator for a panel which we refer to as a **panel coordinator**.

- The coordinators 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.

## Structure of the solution

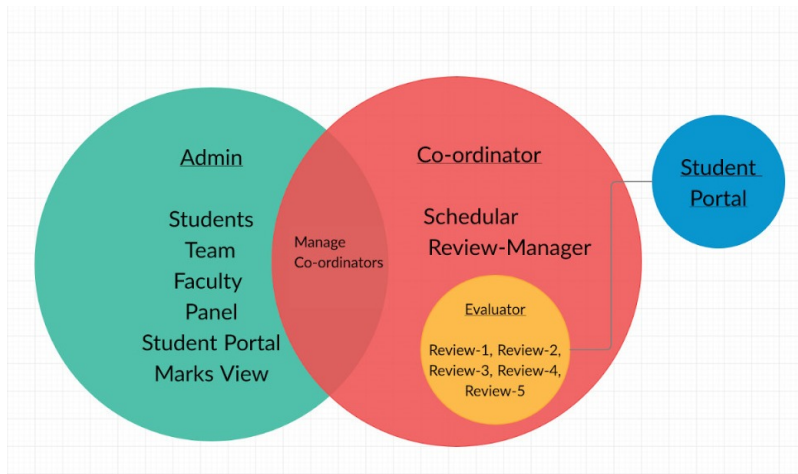


Figure 1: User Roles

1. The application at its core is designed based on a hierarchy. Each user can access and modify the content only within the boundaries set up by their role as described in Figure 1. The 3 Roles are Administrator, Coordinator and Evaluator
2. **Administrator**
  - The administrator is incharge of inserting and maintaing the meta data for students, teams and faculties.
  - The administrator can initialize panels with faculties and teams in it.
  - The administrator also has a few special permissions like opening/closing the student portal, mail student portal passwords to students and download the marks summary for each student.
3. **Coordinator**
  - The coordinator is incharge of setting up project reviews for all the teams in their panel.
  - The coordinator needs to assign a list of faculties for each team for each project review who shall in turn evaluate the team.
  - The coordinator must also set up opening and closing time for each project review within which the evaluators must evaluate the teams.

#### 4. Evaluator

- Evaluators' role is to evaluate all the teams assigned to them for every project review.

#### 5. Student Portal

- Students are given a “mini” user status when they submit their project details to the project coordinator. They can set up their **profile image**, project description as well as view the public comments provided by their reviewers.
- This portal is opened by the administrator.

#### 6. Other Salient Features

- The interface provides *E-Mail support*. eg:
  - The coordinators can mail the review schedule to all the evaluators in the panel.
  - Passwords are generated and mailed to all students to access the student portal.
- The interface also provides *CSV Upload/Download*.
- Evaluators can view profile images submitted by students.

### Implementation

#### 1. REST-API

- The back-end of the web application is a REST API which interacts with clients using JSON requests and responses.
- The backend of the web application is developed using Django Framework.
- PostgreSQL is the database engine used.
- Django ORM is used to convert the relational data (postgres) to objects.

#### 2. Security

- The web application authenticates the user using a combination of **JWT tokens** and **Django Session Cookies**.
- The *JWT Access Token* is used to authenticate the user to communicate 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.

#### 3. Single Page Application

- Implementation of a **Single Page Application** using breadcrumb structure with *vanilla javascript*.
- User Interface is designed with **Bootstrap v4.5**.

#### 5. Dockerized for cross-platform reliability

- The web application is **dockerized** for reliability across platforms.

- To reduce the hassle of installing third party libraries and easy deployment, the app is dockerized into containers for the *REST API*, *Nginx*, *PostgreSQL*

## Requirements

Packages	Version	Details
PostgreSQL	12.4	Database Used With Django
Python	3.6	
psycopg2-binary	2.8.5	PostgreSQL API for python
Django	3.0.8	Python Framework
django-bleach	0.6.1	To Prevent XSS attcks
djangoRESTframework	3.11.0	REST API
djangoRESTframework-simplejwt	4.4.0	For JWT Tokens
Pillow	7.2.0	Image Processing
uWSGI	2.0.19.1	To Deploy Django With Nginx
Docker	19.03.6	
docker-compose	1.17.1	Combine Containers
Gunicorn	19.7.1	Python WSGI HTTP Server
Nginx	1.14.0	Load Balancer for HTTP server

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

They can be downloaded by running

```
sudo apt install docker docker-compose nginx gunicorn
pip install -r requirements.txt
```

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 docker-compose file is as follows The number of workers have been set to 2, they can be changed accordingly here.

```
version: "3"

services:
  db:
    image: postgres
    container_name: nvrD-postgres
```

```

volumes:
  - postgres_data:/var/lib/postgresql/data/
environment:
  - POSTGRES_USER=postgres
  - POSTGRES_PASSWORD=postgres
  - POSTGRES_DB=nvrdb
web:
  build: .
  command: bash -c "cd Back_end/NVRD && python3 manage.py makemigrations && python3
    manage.py migrate && gunicorn --workers=2 NVRD.wsgi:application --bind 0.0.0.0:8000"
  container_name: nvr-django
  env_file:
    - .env.dev
  volumes:
    - ./code
  ports:
    - "8000:8000"

nginx:
  build: ./nginx
  container_name: nvr-d-nginx
  ports:
    - 80:80
  depends_on:
    - web

volumes:
  postgres_data:

```

Run the following commands to start the server in the project directory. Choose 0 when prompted to initialize the database with an admin user.

```
bash run.sh
```

run.sh file contains the following code

```

read -p "enter 1 to initialize database (default NO): " dbinit
sudo docker-compose up --no-start
sudo docker-compose start
echo "NVRD containers are running"
dbinit=$dbinit
if ((dbinit==1)); then
  sudo docker exec nvr-d-jango bash -c "python Back_end/NVRD/manage.py flush --noinput"
  sudo docker exec nvr-d-jango bash -c "python Back_end/NVRD/manage.py shell
    < Back_end/NVRD/eval/values.py"
  echo "nvrdb initialised"
fi

```

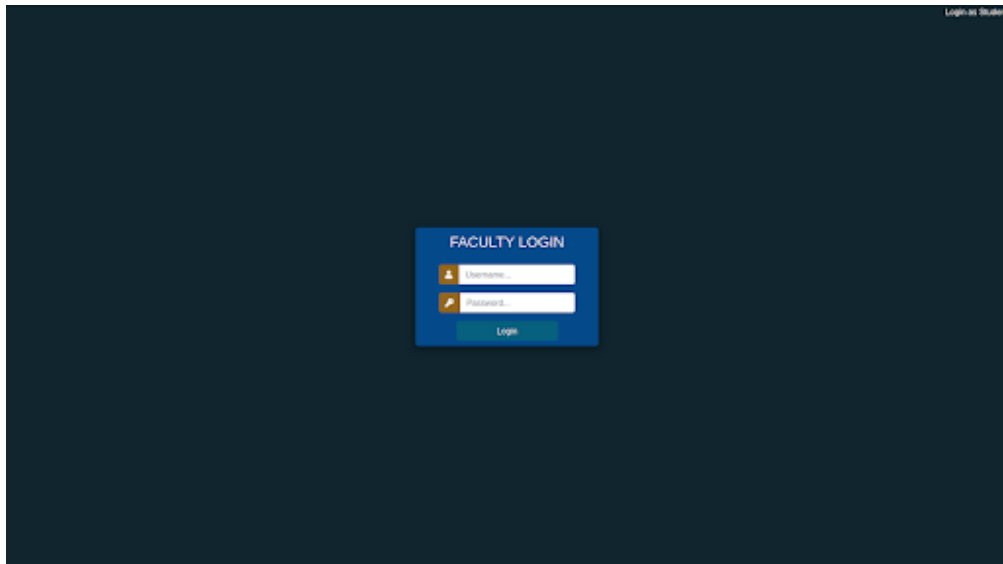
## The Actual Website

The following are some images of our website.

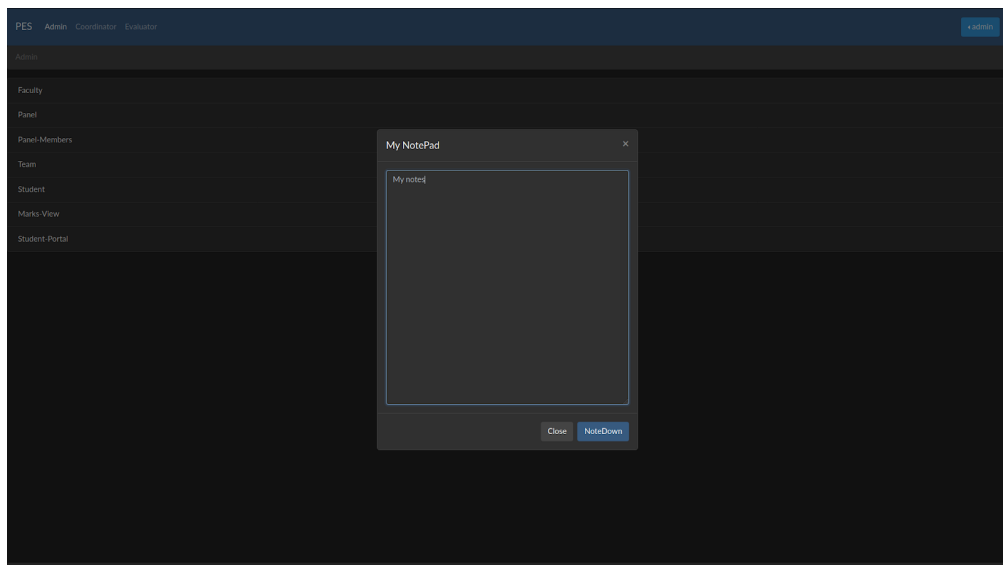
---

### Images

---



Login page for faculty



Faculty Home Page

Images

PESAdminCoordinatorEvaluator

AdminMarks View

SRN

PES1201802018

PES1201802017

PES1201802016

PES1201802015

PES1201802014

PES1201802013

PES1201802012

PES1201802011

PES1201802010

PES1201802009

PES1201802008

PES1201802007

PES1201802006

PES1201802005

PES1201802004

Result

PES1201802006

goodness7

Review 1

Concept Of The Work (10)	Methodology Proposed (10)	Literature Survey (10)	Knowledge On The Project (10)	Is Evaluated	Comments	Faculty ID	Faculty Name	Faculty Type
0	0	0	0	✗	not yet scored	admin	admin	Professor
0	0	0	0	✗	null	eval1	eval1	Professor
Review 2								
Review 3								
Review 4								
Review 5								
Total								

goodness7

Tim\_2020

0000000007

CSE

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

||||XXXX

XXXXXX

XXXXXX

Status

Student Portal closed

Marks View

PESAdminCoordinatorEvaluator

AdminStudent Portal

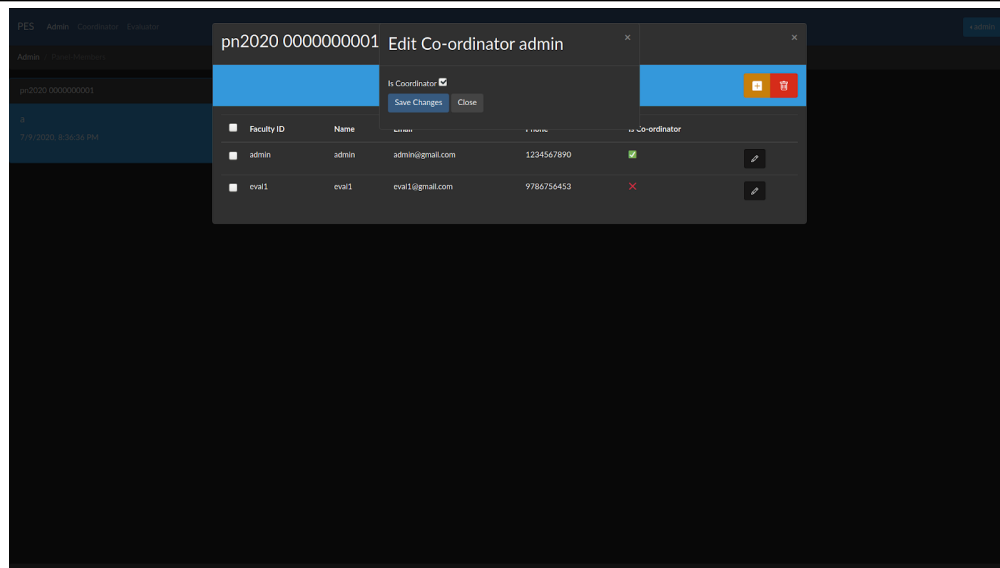
STUDENT PORTAL

Status

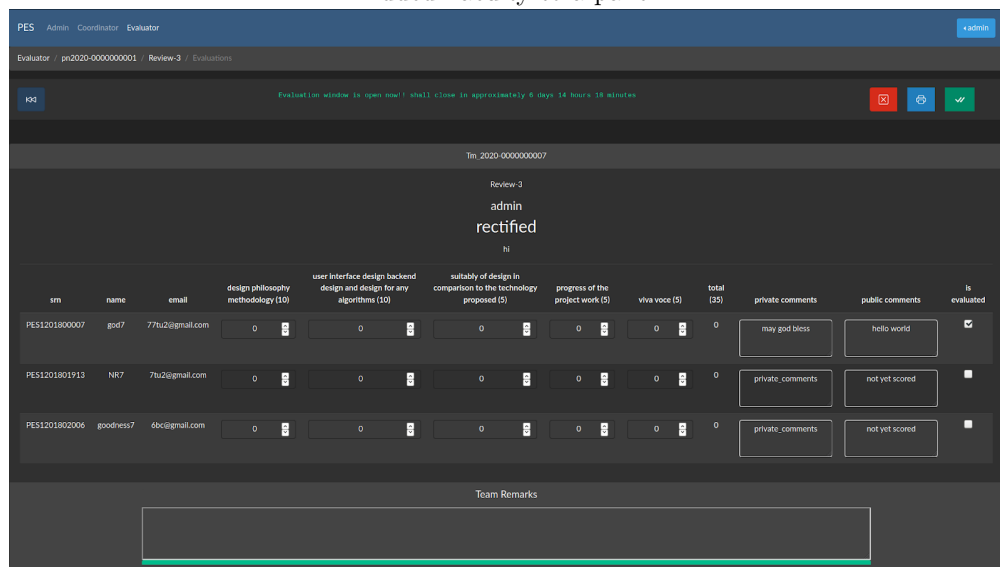
Student Portal closed

Student Portal

## Images



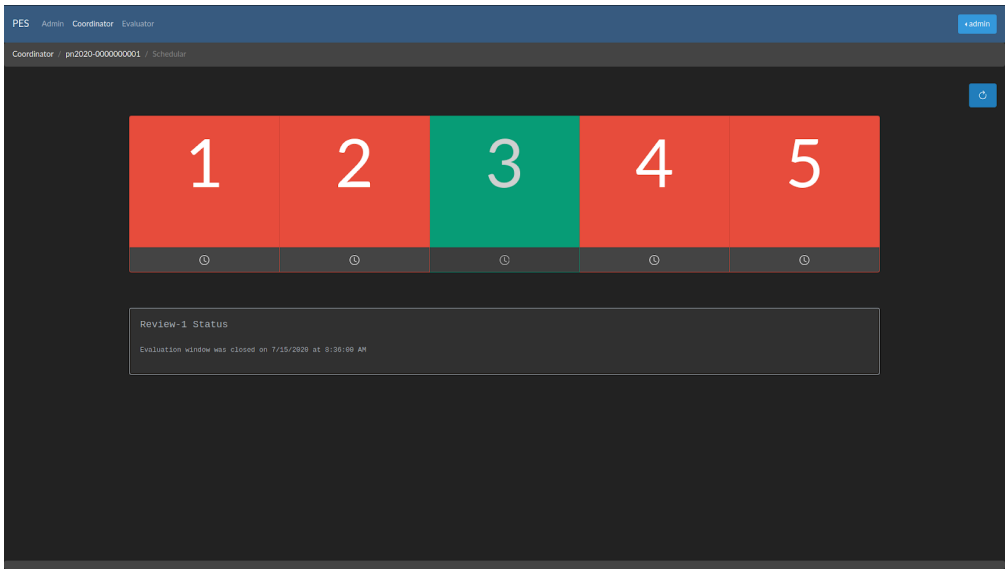
## Added Faculty to a panel



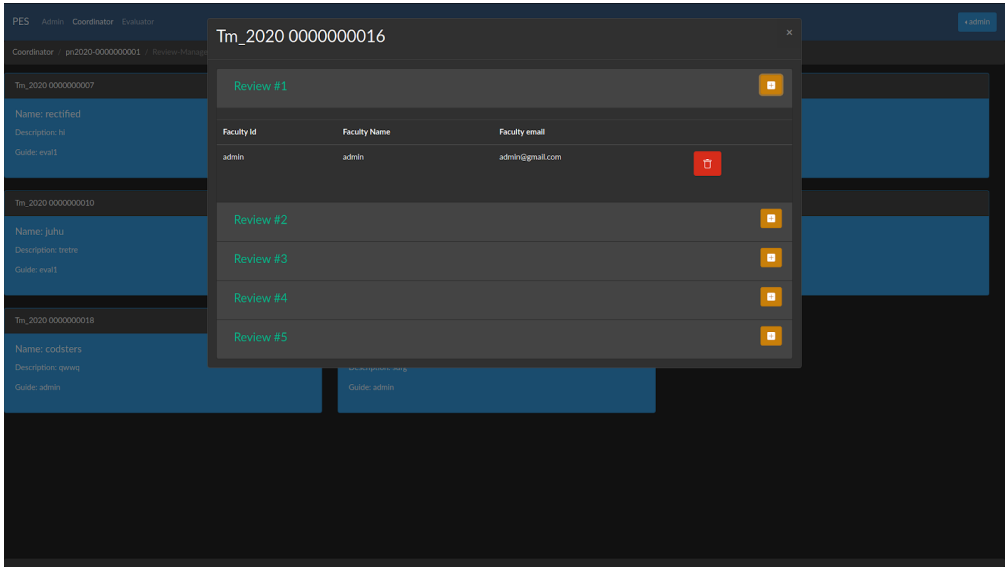
## Review Form



Images



Review Scheduler for Panel Coordinators



Evaluator Assigning Faculty for reviews