



**BITS Pilani, K. K. Birla Goa Campus**

**Project Report**

**Course Load Portal**

Submitted in partial fulfillment of the course

**CS F266**

Submitted To:

**Prof. Neena Goveas**

Submitted by:

**ADARSH NANDANWAR**

**2018A7PS0396G**

# TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b>	<b>2</b>
<b>COURSE LOAD PORTAL</b>	<b>3</b>
INTRODUCTION	3
<b>FEATURES</b>	<b>4</b>
<b>UPDATE 1</b>	<b>5</b>
TECHNOLOGY STACK	6
FRONT END	6
BACK END	6
<b>SOURCE CODE</b>	<b>7</b>
<b>MANUAL</b>	<b>9</b>
LOGIN	9
ADMIN PAGE	10
DASHBOARD	11
FIRST PANEL	12
SECOND PANEL	13
THIRD PANEL	14
TOP PANEL	14
<b>CONTRIBUTION</b>	<b>15</b>

# **COURSE LOAD PORTAL**

## **INTRODUCTION**

The main aim of this project is to automate the course load generation process for every semester for the smooth functioning of the AUGSD. This software replaces the manual distribution of the excel sheets to HoDs and manual generation of the course load in the required formats. The HoD can now fill the data of the courses offered in the next semester easily in a responsive user interface. This data can then be downloaded in different formats by the HoDs and admin as explained below. There is also an option to upload comment files for extra information by the HoD.

The Project was built on Django (Python), React.js, and PostgreSQL. The web application was hosted on Heroku server and the file storage on Amazon AWS S3.

Link: <https://augsd-course-load.herokuapp.com/>

Source code: <https://github.com/AdarshNandanwar/augsd-course-load>

# FEATURES

1. Accounts for every department in the Institute
2. HoD Side:
  - a. Data collection via an online dashboard page. Here the respective HoDs of the departments can input the data using the responsive user-friendly UI.
  - b. Option to add comment file for extra information
  - c. Download the course load in 2 formats for their department (The HoD can view only the course load of their department). The formats are:
    - i. Course wise
    - ii. Instructor wise
  - d. Erase the data for a particular course
  - e. Add/modify the data of other department courses by requesting the course access in the add tab.
3. Admin Side:
  - a. The admin page offers very powerful tools to manipulate the database.
  - b. Here the admin can download all the comment files sent by the HoDs.
  - c. Edit the course and instructors details in the database:
    - i. Add
    - ii. Modify
    - iii. Delete
  - d. Download the course load in 4 formats with a single click
    - i. Course wise
    - ii. Instructor wise
    - iii. ERP
    - iv. Time Table
  - e. The admin can upload the initial data in the format of the provided data\_template before every semester. This includes courses (CDCs and electives), instructors (faculty and Ph.D. students), and department mapping.
4. The application can also handle the equivalent courses whose data is supposed to be common.
5. The courses which are common between departments can be seen and modified by both.

# UPDATE 1

1. Major changes in database models
2. Refactored multiple pre-existing APIs including APIs to download course loads, `get_data`, `get_course_data`, etc. to make it work with the new database models.
3. Developed new APIs for the course history feature.
4. Added ME TAs in the database model with all the related functions for them.
5. Minor bug fixes
6. HoD Dashboard:
  - a. The courses not to be offered in the semester can be disabled easily without loss of data now using a toggle button.
  - b. The past course strength of every course is now displayed to the HODs for them to make informed decisions regarding the capacities of the courses.
  - c. On downloading the course load from HOD Dashboard, only the enabled courses are now visible in all the formats.
  - d. Previous year data is now displayed to the HODs including the course IC, course instructors, number of sections, etc.
  - e. All courses are not displayed now. Only those which can be offered in the current sem (odd/even) are displayed.
7. Admin Side:
  - a. Added controls to turn on/off the portal for all HOD accounts
  - b. Added option to upload the past course capacities directly to the database.
  - c. The admin can now view all the changes in seats made by the HODs for every course over time.
  - d. Added new course load download format:
    - i. Compressed Instructor wise format
  - e. Minor changes in the following output format:
    - i. ERP
    - ii. Time Table

# TECHNOLOGY STACK

## FRONT END

- React.js Framework - Responsive User Interface (HOD Dashboard)
- Django template language (Admin Panel)
- JavaScript, jQuery
- HTML, CSS, Bootstrap 4

## BACK END

- Django Framework - Python-based backend framework
- PostgreSQL - Database
- Heroku - Deploying and testing
- Amazon AWS S3 - Persistent file storage server

# SOURCE CODE

Source code: <https://github.com/AdarshNandanwar/augsd-course-load>

The root directory consist of three applications:

- **AUGSD\_time\_table\_project** - root application of the project.
- **course\_load** - the application that handles all the course load related requests.
- **client** - application for React.js responsive user interface.

The root directory also contains:

- **mediafiles** - Contains all the media files uploaded/downloaded by the server.
- **staticfiles** - Contains all the static files (CSS, JavaScript) in one place for Heroku deployment.

📁 AUGSD\_time\_table\_project

📁 client

📁 course\_load

📁 mediafiles

📁 staticfiles

📄 .gitignore

📄 Procfile

📄 README.md

📄 data.xlsx

📄 data\_template.xlsx

📄 manage.py

📄 package.json

📄 populate.py

📄 requirements.txt

📄 yarn.lock

The purpose of some of the files that are there inside the course\_load application have been explained below:

Scripts:

- **forms.py** - Form data is stored inside the application directory. Form class manages form data and performs validation when a form is submitted.
- **models.py** - A model is a class that represents a table or collection in the database, where every attribute of the class is a field of the table or collection. It contains the essential fields and behaviors of the data we need to store.
- **urls.py** - The path function is contained within the urls.py. Path function is used for routing URLs to the appropriate view functions within a Django application using the URL dispatcher.
- **utils.py** - This file contains the code for other essential functions such as importing the data from the given input file, populating the database, and other functions to help in the smooth functioning of the application.

📁 migrations

📁 templates

📁 views

📄 admin.py

📄 apps.py

📄 forms.py

📄 models.py

📄 tests.py

📄 urls.py

📄 utils.py

Directories:

- **views directory** - This directory contains *admin\_views.py*, *hod\_views.py*, *shared\_views.py*. These contain view functions. A view function is a Python function that takes a web request and returns a web response. This response can be the HTML contents of a Web page, or a redirect, or a JSON file, or a 404 error, or an XML document, or an image, anything that a web browser can display.
- **templates directory** - This directory contains all static web pages for admin, course\_load, and registration required for frontend.

📄 \_\_init\_\_.py

📄 admin\_views.py

📄 hod\_views.py

📄 shared\_views.py

📁 admin

📁 course\_load

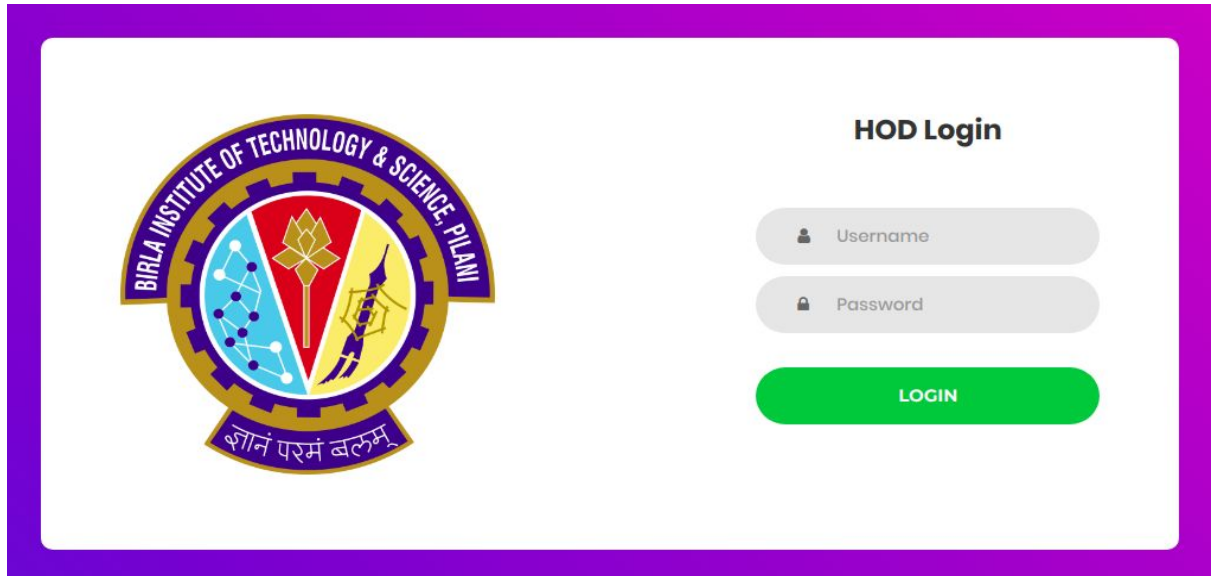
📁 registration

Source code: <https://github.com/AdarshNandanwar/augsd-course-load>



# MANUAL

## LOGIN



The screenshot shows the HOD Login page for the Birla Institute of Technology & Science (PILANI). On the left is the institute's logo, which is a circular emblem with a gear-like border. Inside the circle, there is a red lotus flower in the center, a blue molecular structure on the left, and a yellow book with a pen on the right. The text "BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI" is written around the top of the circle, and "ज्ञानं परमं बलम्" is written at the bottom. On the right side of the page, the title "HOD Login" is displayed. Below the title are two input fields: "Username" with a person icon and "Password" with a lock icon. At the bottom of the login form is a green button labeled "LOGIN".

The HoDs of the department and admin can log in to the web portal using the login page. Each department HoD has been assigned a different username and password using which they can log in to the web portal.

## ADMIN PAGE

If the admin logs in, the page will be redirected to the admin page. Here the admin can download the files uploaded by the HoDs, download automatically generated course load in 4 different formats, and add, modify, or delete the course and instructors in the database. The initial data for the semester can be uploaded by the “Upload data” button.

The Admin Page interface is displayed within a purple border. It features a central white area with the title "Admin Page" at the top. On the left, under "Download HOD comments:", there is a list of departments: BIO, CHE, CHEM, CS, ECON, EEE, HUM, MATH, MECH, and PHY. On the right, there are three sections of buttons: "Add:" with "Course" and "Instructor" buttons; "Update:" with "Course" and "Instructor" buttons; and "Delete:" with "Course" and "Instructor" buttons. In the center, under "Download course load:", there are five blue buttons: "Course-wise", "Instructor-wise", "ERP", "Time Table", and "Format 5". At the bottom, under "Other Controls:", there are four dark grey buttons: "Initialize database", "Upload course capacity", "Course history", and "Activate portal".

# DASHBOARD

After the HoD has successfully logged in, the following page will appear.

AUGSD Course Load

DOWNLOAD COURSE-WISE

DOWNLOAD INSTRUCTOR-WISE

UPLOAD COMMENT FILE

CLEAR COURSE DATA

SUBMIT

CDC

ELECTIVES

ADD

EQUIVALENT COURSE

CS F111 (COMPUTER PROGRAMMING)

CS F211 (DATA STRUCTURES & ALGORITHMS)

CS F212 (DATABASE SYSTEMS)

CS F213 (OBJECT ORIENTED PROG)

CS F214 (LOGIC IN COMPUTER SC)

CS F222 (DISCR STRUC FOR COMP SCI)

CS F301 (PRINCIPLES OF PROGG LANG)

CS F303 (COMPUTER NETWORKS)

CS F342 (COMPUTER ARCHITECTURE)

CS F351 (THEORY OF COMPUTATION)

CS F363 (COMPILER CONSTRUCTION)

CS F364 (DESIGN & ANAL OF ALGO)

COURSE INFO

NO. OF LECTURES

0

NO. OF TUTORIALS

0

NO. OF PRACTICALS

0

IC

MAX STRENGTH PER LECT

0

MAX STRENGTH PER TUT

0

MAX STRENGTH PER PRAC

0

PAST COURSE STRENGTH

-1

COURSE ACTIVE

FACULTY INFO

LECTURES

ADD FACULTY

TUTORIALS

ADD FACULTY

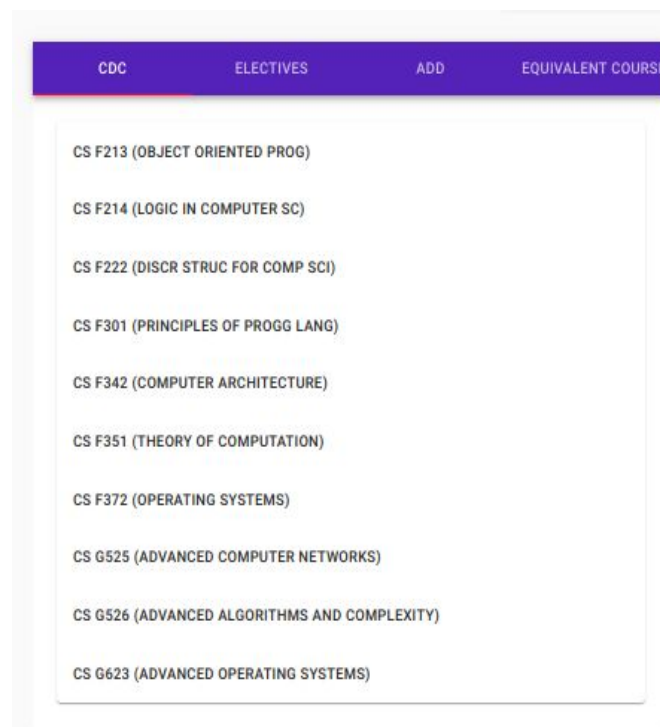
PRACTICALS

ADD FACULTY

## FIRST PANEL

In the dashboard, the left panel shows four headings:

- **CDC** - All compulsory courses for the semester from the department are listed here.
- **ELECTIVES** - All electives and WILP courses from the department are listed here.
- **ADD** - All other courses not in the CDC or ELECTIVES list can be added from here.
- **EQUIVALENT COURSES**: Courses equivalent to the selected course are listed here.



If the user wants to add a new course to the course list that is currently not present in it, the user can click on ADD and select a course that the user wishes to add. The selected course will get added to the CDCs or ELECTIVE depending on the type of course.



If the user selects a course in CDC or ELECTIVES tab and clicks on the EQUIVALENT COURSES tab, equivalent courses for the course will be displayed.



If the details for anyone of the equivalent courses is filled, the same data will be automatically filled in the other equivalent courses too.

## SECOND PANEL

When the user clicks on any course from CDC or ELECTIVES tab, the selected course will appear in the second panel “COURSE INFO”.

The number of sections submitted in the previous semester will appear under “No. of Lectures”, “No. of Tutorials” and “No. of Practicals”.

This number can be modified for the current semester if required. The user can select the IC, type the max strength per section for the lecture, tutorial, and practical. Finally, activate the course.

A screenshot of the 'COURSE INFO' panel. It features several input fields and a toggle switch. The fields are labeled: 'NO. OF LECTURES' (value 0), 'NO. OF TUTORIALS' (value 0), 'NO. OF PRACTICALS' (value 0), 'IC' (dropdown menu showing 'IC'), 'MAX STRENGTH PER LECT' (value 0), 'MAX STRENGTH PER TUT' (value 0), 'MAX STRENGTH PER PRAC' (value 0), and 'PAST COURSE STRENGTH' (value -1). At the bottom, there is a 'COURSE ACTIVE' label and a toggle switch that is currently turned off.

## THIRD PANEL

Here, the user can add instructors in the sections for the course. The user needs to specify the section number from the dropdown 'SEC'. The user can click on the 'x' sign to remove an instructor if not required. At least one instructor should be assigned to each section as specified under the "no. of sections" in the "course info" panel.

Any number of instructors can be assigned against each section. Instructor dropdown will give names of faculty and Ph.D. students. After filling up the faculty details for all sections, the user can click on SUBMIT on the top panel to save. If incomplete information is given it will not be allowed to submit. Once submitted the course will be highlighted in the course list. To make any changes, users can click on the course again and incorporate necessary changes and submit them again.

The screenshot shows a panel titled 'FACULTY INFO'. It contains three sections: 'LECTURES', 'TUTORIALS', and 'PRACTICALS'. Each section has a table with columns for 'LECTURE' and 'SEC'. In the 'LECTURES' section, the first row has 'NEENA G...' in the 'LECTURE' column and 'SEC' in the 'SEC' column, with a red 'x' icon to the right. The second row is empty. Below the table is a red button labeled 'ADD FACULTY'. The 'TUTORIALS' and 'PRACTICALS' sections have similar empty tables and red 'ADD FACULTY' buttons.

## TOP PANEL

The screenshot shows a panel with five buttons. At the top are two red buttons: 'DOWNLOAD COURSE-WISE' and 'DOWNLOAD INSTRUCTOR-WISE'. Below them is a red button labeled 'UPLOAD COMMENT FILE'. At the bottom are two buttons: a red button labeled 'CLEAR COURSE DATA' and a blue button labeled 'SUBMIT'.

**CLEAR COURSE DATA** - Deletes the course data.

**DOWNLOAD COURSE WISE/ DOWNLOAD INSTRUCTOR WISE** - Downloads the data entered at any point in time in the corresponding format.

**UPLOAD COMMENT FILE** - Let users upload a file if additional information needs to be given.

# CONTRIBUTION

The project was made by a team of 3-4 members. My contribution to the project:

- Point of contact for project development between AUGSD and the software team.
- Solely responsible for the back end of the whole project which was built on Django.
  - All the APIs in the application were written by me.
- Handling of the database. The database used here is PostgreSQL.
  - All the queries were written by me.
- All the file handling in the software involving:
  - Populating database before each semester from an excel file.
  - Upload and download of comment file by the HoDs and admin.
  - Generation of customized course load which is different for different users in multiple formats.
- The front end of the admin side and all its functionalities.
- Comment pages in both the admin and HoD dashboard.
- Setup of persistent storage server using Amazon AWS S3.
- Responsible for hosting the web application on Heroku and AWS after every update on the application for use by the AUGSD.

Team members:

**Adarsh (Me)** - Backend and frontend (admin side)

**Ishant** - Frontend (HOD dashboard)

**Nilesh** - Frontend (HOD dashboard)