WEEKLY OVERVIEW OF INTERNSHIP ACTIVITY

SL.NO.	WEEKS	TOPICS COVERED
1.	Week 1	Outline of Internship and concepts of python programming language followed by assignments.
2.	Week 2	Overview of Angular application architecture followed by configuration of Angular application and understanding its components.
3.	Week 3	Working with MongoDB and Robo 3T and performing database queries.
4.	Week 4	Client-server communication using both Angular HTTP module and the REST API where APIs were built using flask module. Interfacing of PyMongo and MongoDB were also covered.
5.	Week 5	Project implementation.
6.	Week 6	Project implementation.

CHAPTER 1

INTRODUCTION

The Internship was conducted by Heraizen Technologies Pvt. Ltd. located at Bangalore. The internship focused on a variety of topics related to web including database management using NoSQL. Angular framework was used as a medium for application development. Various features of Web development were explored throughout the internship.

The internship program focused on the following topics:

- Setting up the Development Environment and working with source code repositories (Visual Studio Code, Github)
- Fluent understanding of Python
- Working with NoSQL DB and PyMongo
- Working with RESTFul Web Services using Flask
- Developing webpages using Angular with Bootstrap
- Understanding the process of Requirements Gathering, Design and Implementing the project while collaborating with the team
- Documentation and Presentation of project

OBJECTIVE

The main objective of this internship was to give a complete guide to Angular and how it can be used to develop a single page web application. It also focused on how the application can be connected to the MongoDB database and fetch the data using PyMongo Python module.

CHAPTER 2

ABOUT THE COMPANY

The current chapter gives an elaborate view about the company, Heraizen Technologies and Pvt. Ltd. The summary of the company, services provided by the organization are presented in this chapter.

2.1 SUMMARY OF THE COMPANY

Heraizen Technologies Private Limited[1] is an unlisted private company incorporated on 20 April, 2017. It is classified as a private limited company and is located in Bangalore, Karnataka. It is the trusted, digital transformation partner of choice for education campuses, universities, institutes of higher learning, schools and educational institutes of all kinds. It is a strongly customer-focused, engineering-driven organization with deep commitment to education domain. Their product innovations come from market insights arising from in-depth customer discussions that they pursue relentlessly.

2.2 SERVICES PROVIDED

Heraizen Technologies Private Limited involved in Software publishing, consultancy and supply.

- Software publishing: It includes production, supply and documentation of ready-made (non-customized) software, operating systems software, business and other applications software, computer games software for all platforms.
- Consultancy: It includes providing the best solution in the form of custom software after analyzing the user's needs and problems.
- Custom software: It includes made-to-order software based on orders from specific users.

CHAPTER 3

TECHNOLOGIES COVERED

3.1 PYTHON

Python is a widely used general-purpose, high level programming language. It is used for web development (server-side), software development, mathematics, system scripting. Python provides web frameworks which are utilized in the backend for server-side technology, aiding in URL routing, HTTP requests and responses, accessing databases, and web security. While it is not required to use a web framework, it is extremely recommended because it helps you develop complex applications in significantly less time[2].

3.2 MONGODB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, it uses JSON-like documents with optional schemas. A record in MongoDB is a document, which is a data structure composed of field and value pairs. The values of fields may include other documents, arrays, and arrays of documents. MongoDB stores documents in collections. Collections are analogous to tables in relational databases. MongoDB provides high performance data persistence[3].

3.3 PYMONGO

The PyMongo distribution contains tools for interacting with MongoDB database from Python. The bson package is an implementation of the BSON format for Python. The pymongo package is a native Python driver for MongoDB. The gridfs package is a gridfs implementation on top of pymongo. using pymongo package in python helps us to configure with a mongoDB database and retrieve documents from the same.

3.4 RESTful APIs

A RESTful API, also referred to as a RESTful web service, is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage. An API for a website is code that allows two software programs to communicate with each another. The API spells out the

proper way for a developer to write a program requesting services from an operating system or other application. A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from scratch.

3.5 FLASK

Flask is a popular Python web framework that is used for developing web applications. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It will act as the central configuration object for the entire application. It is used to set up pieces of the application required for extended functionality, e.g., a database connection and help with authentication. It is regularly used to set up the routes that will become the application's points of interaction[4].

3.6 ANGULAR

Angular is a TypeScript based on open-source web application framework, application design framework and development platform for creating efficient and sophisticated single-page apps. The structure of Angular is based on the components/services architecture. The framework consists of several libraries, some of them core and some optional. You write Angular applications by composing HTML *templates*, writing *component* classes to manage those templates, adding application logic in *services*, and boxing components and services in *modules* The eight main building blocks of an Angular application includes Modules, Components, Templates, Metadata, Data binding, Directives, Services, Dependency injection[5].

3.7 TYPESCRIPT

TypeScript is an open source, pure object oriented with classes, interfaces and it is statically typed. It is a primary language for Angular application development. It is a superset of JavaScript with design-time support for type safety and tooling. Browsers can't execute TypeScript directly. Typescript must be "transpiled" into JavaScript using the tsc compiler, which requires some configuration.

3.8 BOOTSTRAP

Bootstrap is a free and open source front end development framework for the creation of websites and web apps. In computers, the word bootstrap means to boot: to load a program into a computer using a much smaller initial program to load in the desired program. It is a combination of HTML, CSS, and JavaScript code designed to help build user interface components. Bootstrap was also programmed to support both HTML5 and CSS3. Bootstrap includes user interface components, layouts and JS tools along with the framework for implementation. Bootstrap's responsive CSS adjusts to phones, tablets, and desktops[6].

CHAPTER 4

TASK PERFORMED

The Internship had weekly catch-up sessions with trainers where the interns had to report progress to their respective trainers. Each week, various technologies were slated to be completed and progress was recorded. A comprehensive report of commencement of the Internship is provided in this chapter.

4.1 WEEK 1

The first week of the Internship focused on introduction to the internship training. Further, the plan for the entire internship was clarified and a flow of activities was slated. The first week further focused on introduction to the python programming language to be used for development.

4.1.1 INTRODUCTION TO PYTHON

The Internship began with a brief introduction into Python. The various syntaxes, conditional statements, loops concepts, data structures and object oriented concepts of python programming language were covered. Assignments were conducted to ensure better understanding of concepts.

4.2 WEEK 2

The second week of the Internship focused on Angular framework. Angular is an open-source web application framework used to build single-page applications.

4.2.1 INTRODUCTION TO ANGULAR

The Internship further focused on development of single page application using Angular framework. Angular architecture overview, configuring the angular application and its components and are covered in the first week. The following command is used to install angular project. Example projects were done to get better understanding of the angular architecture.

```
ng new project_name
```

4.3 WEEK 3

The third week of the Internship focused on exploring the concepts of NoSQL databases with the introduction to MongoDB.

4.3.1 INTRODUCTION TO MONGODB

The Internship further went on to focus on database management system, MongoDB which is based on NoSQL philosophy for storing data. The installation process of MongoDB and Robo 3T 1.3.1 were done in this week. Robo 3T is a free and lightweight GUI for MongoDB. It is a MongoDB management tool which has a shell-centric cross-platform and is supported by JSON. An elaborate view of the document-based data storage of MongoDB was analyzed and understood. Further, basic operations like creation of data models using the MongoDB schema were explored. At its core, MongoDB presents itself as an open source designed for improvement and scaling ease and a document-oriented database. Each record is actually a document in MongoDB. Documents are stored in a JSON-like format, Binary JSON in MongoDB. BSON documents are the objects that contain an ordered list of saved elements. Each element is composed of a field name and a specific type of value.

4.4 WEEK 4

The fourth week of the Internship focused on exploring more concepts of Angular such as angular services, routing and building RESTFUL APIs using Flask module.

4.4.1 INTRODUCTION TO RESTful APIs

An introduction was provided into the various concepts of creating and using RESTful APIs. The sessions began with an elaborate introduction into the Service oriented architecture paradigm where multiple applications could render and obtain services from each other through the internet. A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data. A RESTful API, also referred to as a RESTful web service, is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. A RESTful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction.

4.4.2 INTRODUCTION TO FLASK

The Internship further went on to focus on Flask module. Flask is a widely used micro web framework for creating APIs in Python. It is a simple yet powerful web framework which is designed to get started quick and easy, with the ability to scale up to complex applications. The following command is used to install flask.

```
python –m pip install flask
```

The following is the simple demonstration of flask usage in developing RESTFUL APIs.

```
from flask import Flask

app = Flask(__name__)


@app.route('/hello/', methods=['GET', 'POST'])
def welcome():
    return "Hello World!"if __name__ == '__main__':
    app.run(host='0.0.0.0', port=105)
```

4.4.3 INTRODUCTION TO PYMONGO

An introduction to the PyMongo, a python module was provided in the fourth week of the internship. PyMongo is the official MongoDB Python driver for MongoDB. It contains tools for interacting with MongoDB databases. The following command demonstrates how you can install the latest version of the module using the command line.

```
python –m pip install pymongo
```

The following demonstration shows the PyMongo connection with the MongoDB instance assuming that a MongoDB instance is running on the default host and port.

```
import pymongo


client = pymongo.MongoClient()
client =  pymongo.MongoClient('localhost', 27017)


company = client['company']
employees = local_db.employees
```

4.5 WEEK 5

In the fifth week of the internship, interns were assigned to develop an industry level project. Further, an introduction to project was given. The acquired knowledge through the Internship successfully helped us to build a project.

4.5.1 ABOUT THE PROJECT

The project was the part of the existing application called DHI. It is the Real-Time Fully Integrated Digital platform for Education campuses which tracks student performance and has built-in intelligence and analytics for student centric approach based on the concept of outcome based education.

Problem Statement: Identification of courses (top / bottom n) based on CO attainment scores at the course level, considering common rubrics (C3.2.2 A).

Problem Explained: The CO attainment scores are arrived at based on the performance of the students in various assessments. These include Internal Assessments, Assignments, Co-curricular Activities, Feedback and University Examinations. These assessments are categorized into Direct method and Indirect method, with each contributing a certain percentage towards the final attainment value. The various ways of assessment that constitute the Direct method and the Indirect method, in turn, contribute a specific percentage towards the attainment of that method. The methods of assessment and their contribution towards the final attainment value is configured at a branch level for an institution. The Course Objectives (CO) are mapped against appropriate Program Objectives and against Program Specific Objective (where applicable).

This problem statement consists of three modules : Faculty module, HOD module, Principal module and strives to display the CO attainment scores for specific courses for the chosen academic year and term, while considering common rubrics.

In Fifth week, the faculty module was completed.

Faculty Module:

- Based on the selection criteria (Academic year, Term) entered by the Faculty, all the COs of all the courses that the faculty has taken are displayed on a bar graph. The COs of each course are clustered together along the X-axis, and the length of the bar along the Y-axis indicates the corresponding value of that CO of that course.

- Clicking on any of the CO bars of a course will navigate the user to the next screen. A bar graph is displayed with additional details of all the COs of the selected course. Each CO is represented by 3 bars clustered, along the x-axis, with the length of each bar being a representation of the corresponding CO attainment values. These three bars are for Total Attainment, Direct Attainment and Indirect Attainment score for the respective CO.
- On clicking on any of the clustered bars of a CO of a course, further details of its attainment will be displayed as Accordion table which consist of Direct attainment details and Indirect attainment details.

4.6 WEEK 6

In the sixth week of the internship, implementation of the given project was continued. Along with the web screens, Responsive Web Design (RWD) screens were completed.

In sixth week, the HoD and Principal modules were completed.

HoD Module:
The functionality when a HoD logs in remains the same as that of the Faculty. However, the HoD gets to see the same details but for all courses offered by the department based on the entered selection criteria.

Principal Module:
The functionality when the Principal logs in remains the same as that of the HoD. However, the Principal gets to see the same details but for all courses offered by the institution across all departments for the entered selection criteria. An additional selection criteria is given to principal i.e., Department along with Academic year and Term.

CHAPTER 5

OUTCOME OF THE INTERNSHIP

The project helped a lot in understand the core concept of Angular, connecting MongoDB with PyMongo and creating RESTFUL API's using Flask module.

In this chapter the web screens and the Responsive Web Design (RWD) screens of the project is shown.

5.1 WEB SCREENS

The following are the screenshots of the web screen for Faculty module and web screen design remains same for the HoD and Principal modules.

Screenshot 1: The faculty has to select Academic year and term to search CO attainment details of selected criteria.

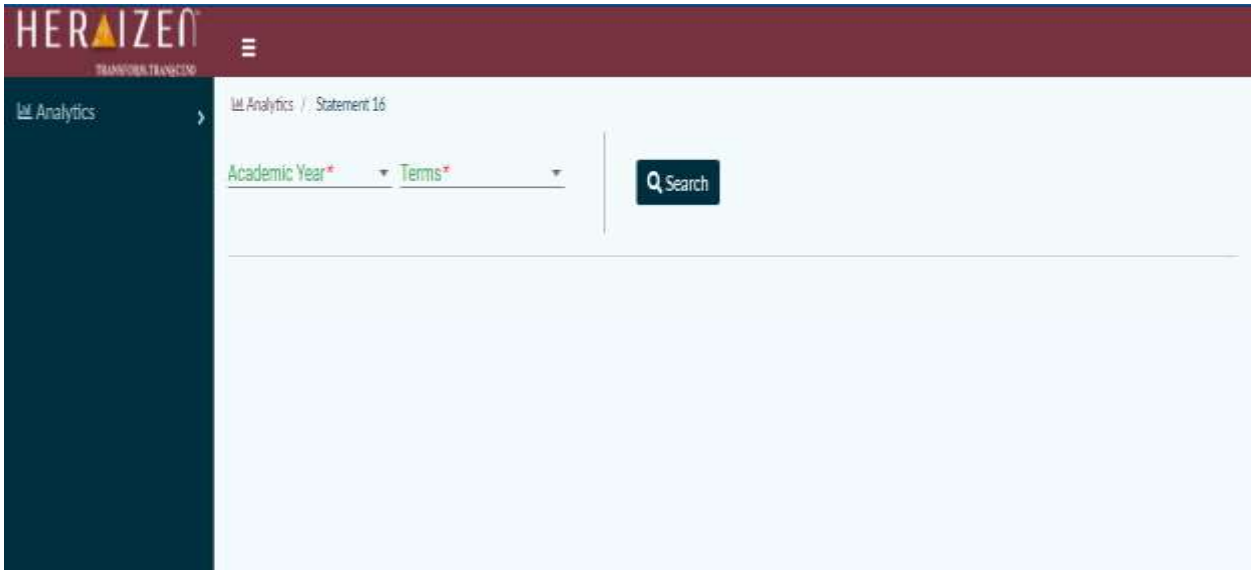

Figure 5.1 RWD of Selection criteria screen for the faculty

Screenshot 2: On selecting particular Academic year and term, faculty can see CO attainment details of all the courses that he/she handles. Hovering on any of the CO bars of a course presents the exact values of the attainment of all the COs of that course in a pop-up window.

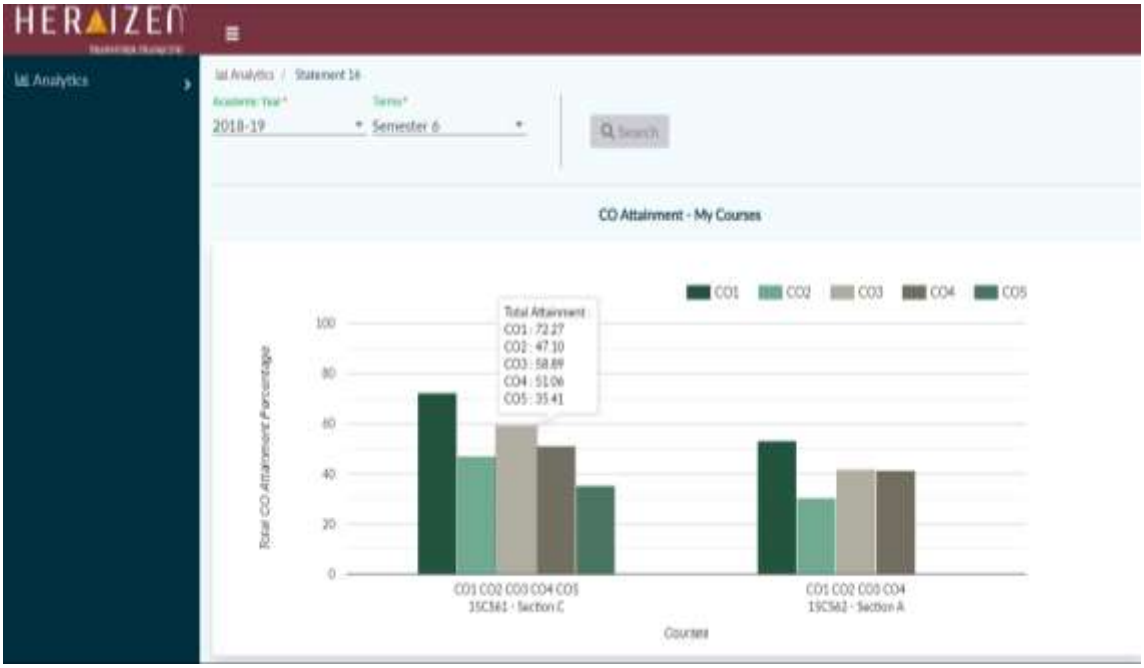

Figure 5.2 RWD of Bar graph of COs of all the courses

Screenshot 3: Clicking on any of the CO bars of a course in the first cluster of the bar graph shown in fig. 5.1.2 gives the detailed information of all the COs of the selected course. Hovering on any of the clustered bars of a CO displays the corresponding three attainment values.

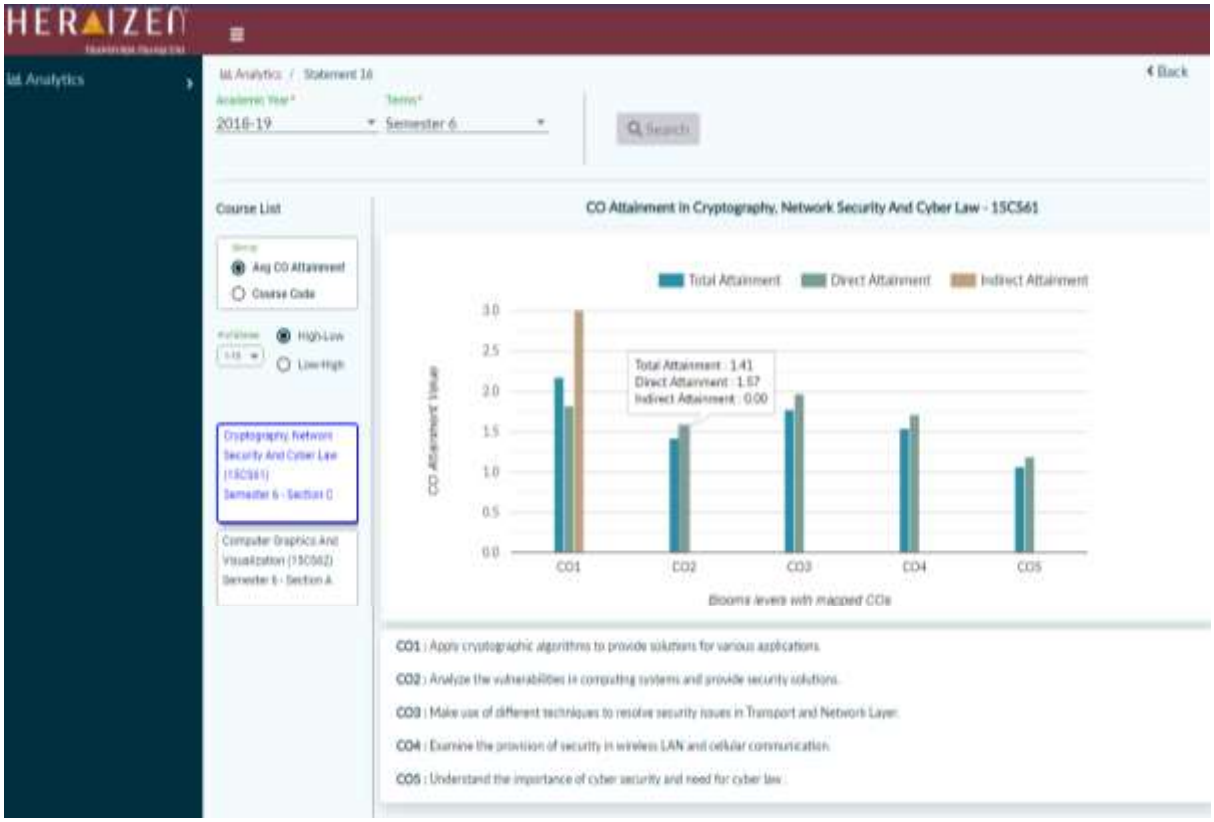

Figure 5.3 RWD of Bar graph showing details of all the COs of the selected course

Screenshot 4: On clicking any of the clustered bars of a CO of a course as shown in fig. 5.1.3 gives the detailed information of selected CO in accordion table.

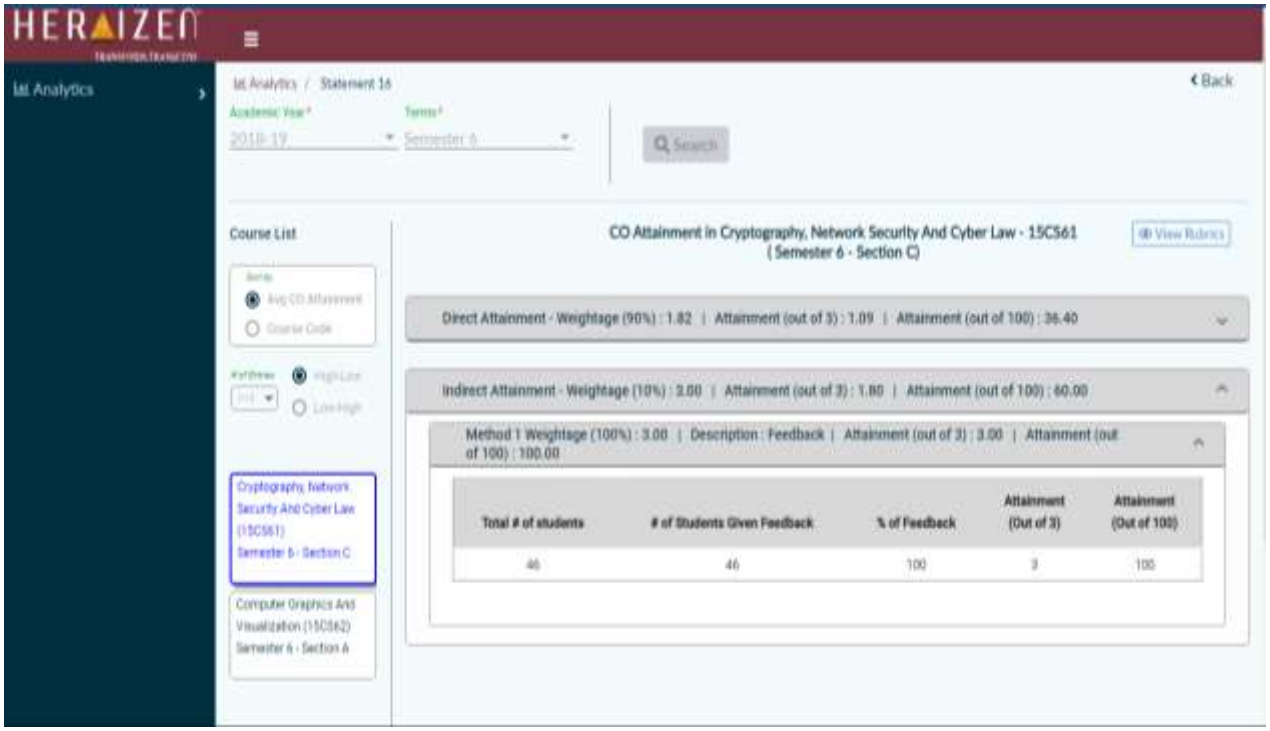

Figure 5.4 RWD of Accordion table showing the detail of selected CO

Screenshot 4: On clicking the View Rubrics button as shown in fig.5.1.4 shows the Rubrics details of the course.

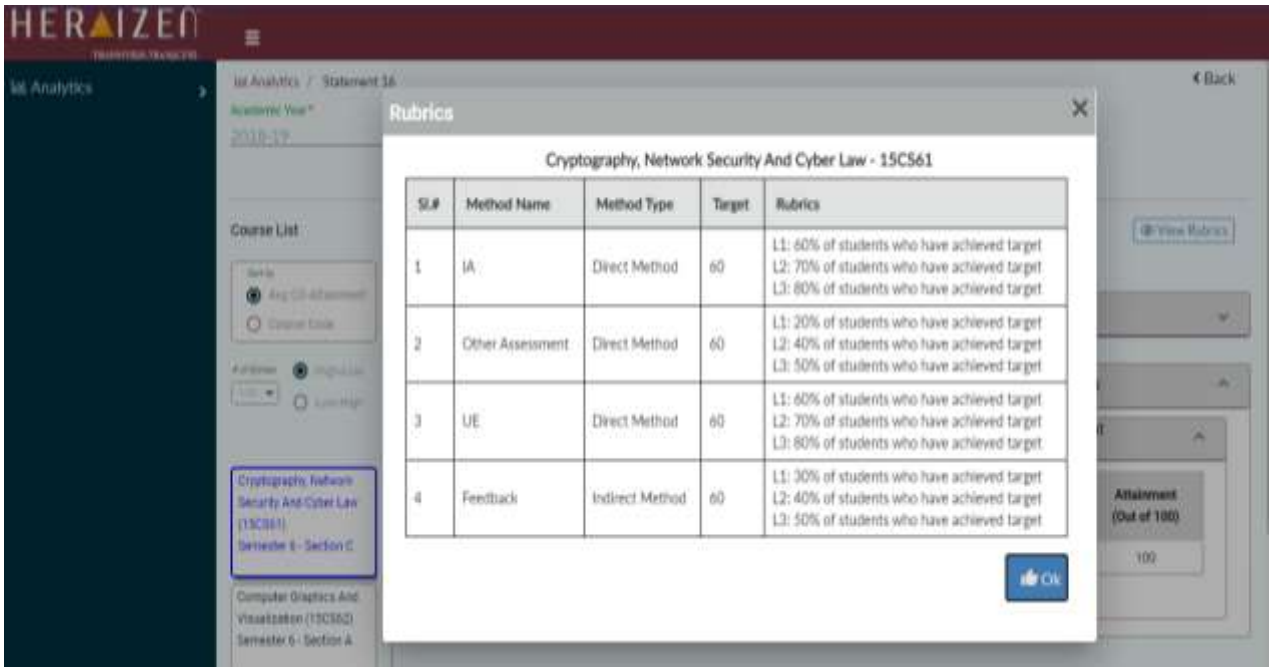

Figure 5.5 RWD of Rubrics details

5.2 RWD SCREENS

Responsive web design (RWD) is a web development approach that creates dynamic changes to the appearance of a website, depending on the screen size and orientation of the device being used to view it.

The following are the screenshots of the RWD screens for Faculty module and RWD screen design remains same for the HoD and Principal modules. The functionality of RWD screens remains same as that of web screens.

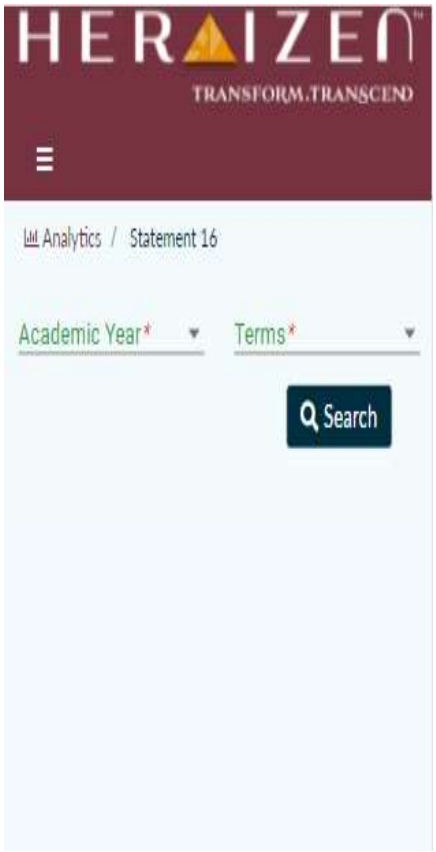

Figure 5.6 RWD of Selection criteria screen for the faculty

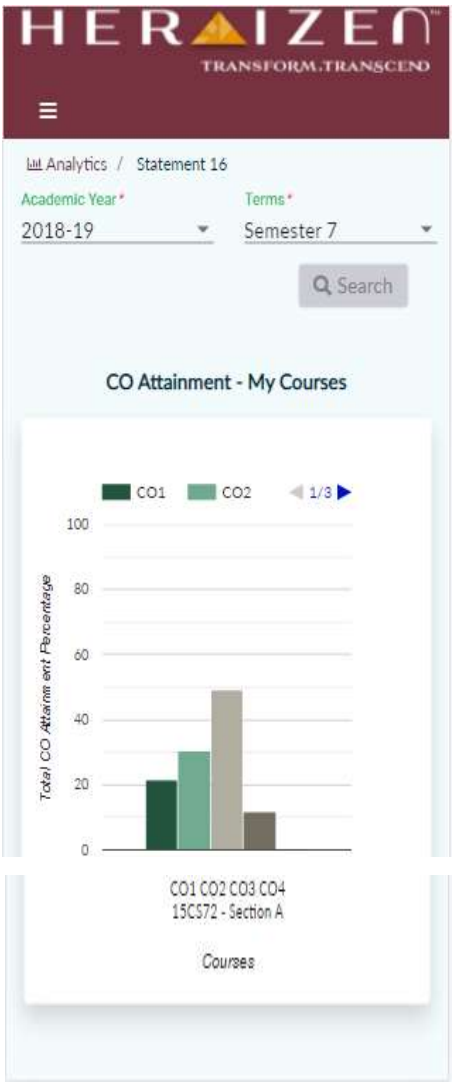

Figure 5.7 RWD of Bar graph of COs of all the courses

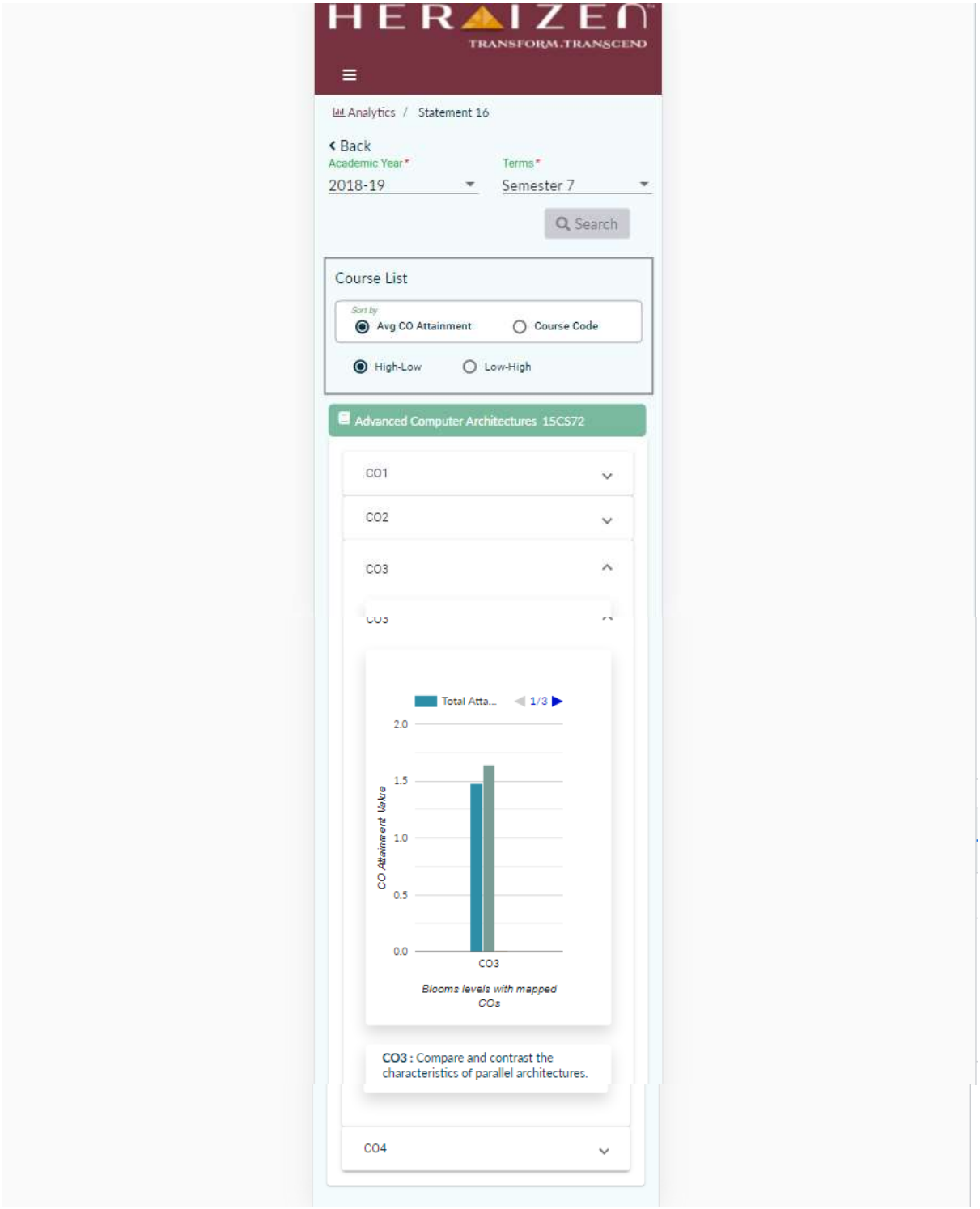

Figure 5.8 RWD of Bar graph showing details of all the COs of the selected course

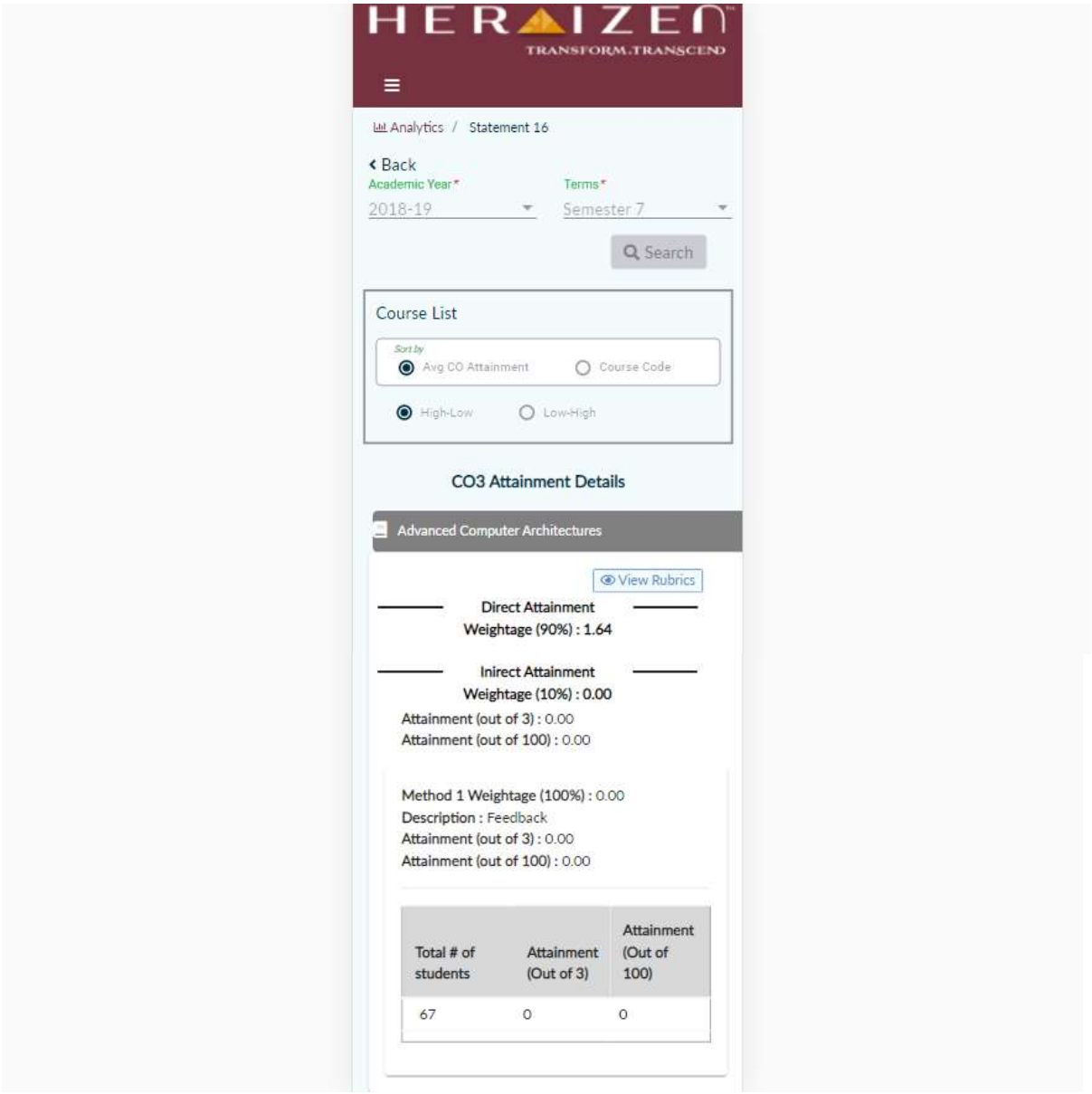

Figure 5.9 RWD of Accordion table showing the detail of selected CO

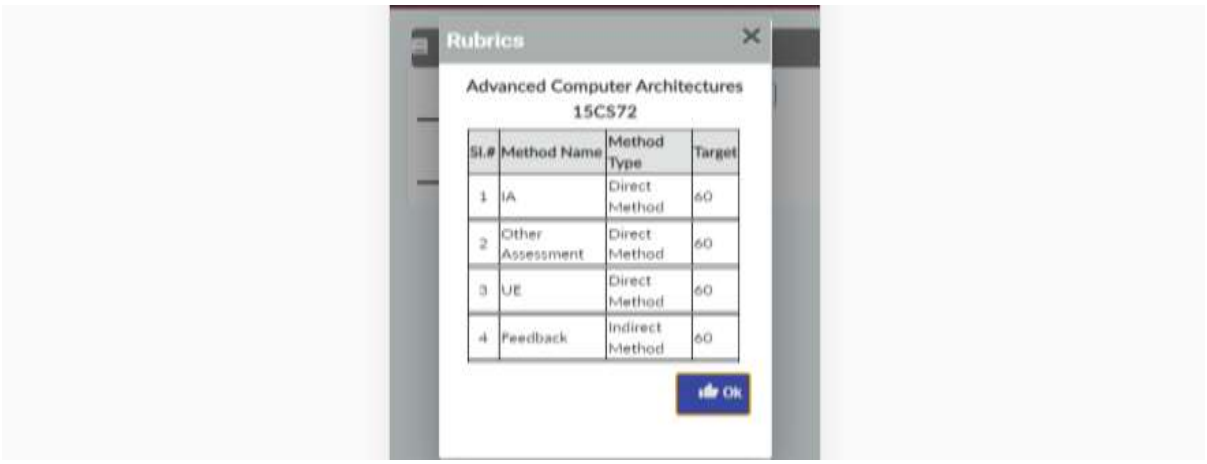

Figure 5.10 RWD of Rubrics details

CHAPTER 6

CONCLUSION

The Internship was successful in providing a clear view into various concepts and methodologies used while developing Web Applications using Angular Framework as the front end framework. It also provided a view into the MongoDB database and the NoSQL paradigm. It also provided knowledge on how the application can be connected to the MongoDB and perform HTTP request and response using flask. Along with theoretical knowledge, having a practice oriented approach is very useful in understanding the concepts which was successfully established in this course.

REFERENCES

[1] https://heraizen.com/

[2] https://docs.python.org/3/contents.html

[3] https://docs.mongodb.com/

[4] https://flask.palletsprojects.com/en/2.0.x/

[5] https://angular.io/docs

[6] https://getbootstrap.com/

[7] https://material.angular.io/