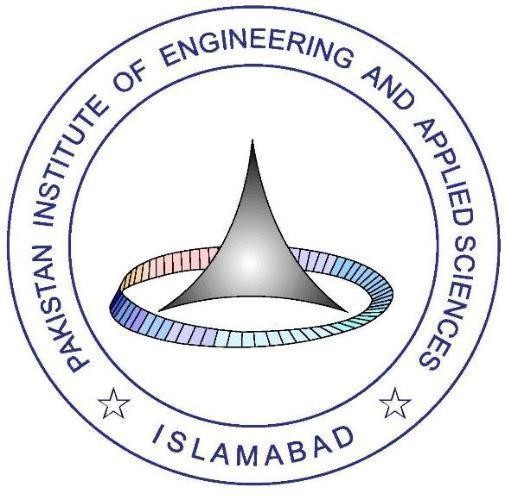
**Development of**

**Job Recruitment Portal**



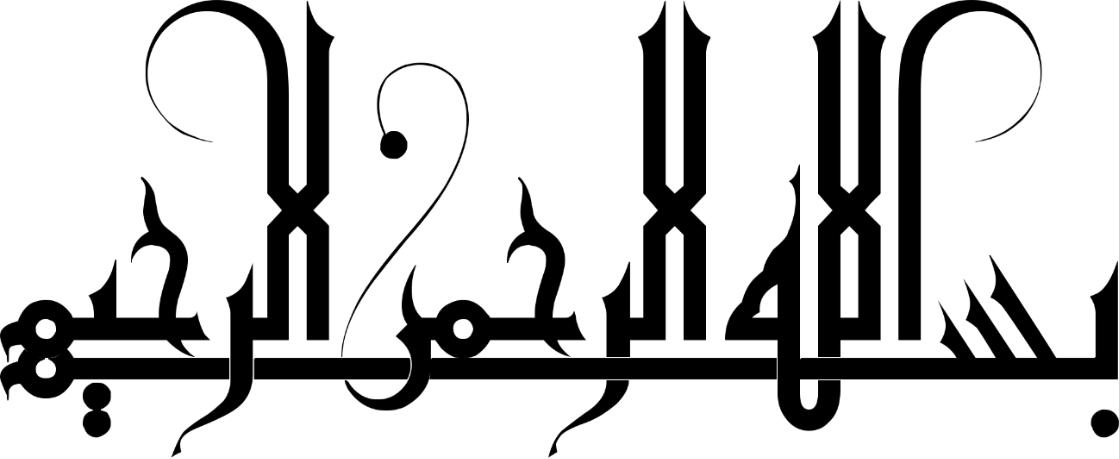
**Mohammad Ahsan Junaid**

**Abdul Rehman**

**Thesis submitted in partial fulfillment of requirements for the Degree of Bachelor of Sciences in Computer and Information Sciences**

**Department of Computer and Information Sciences, Pakistan Institute of Engineering & Applied Sciences, Nilore, Islamabad, Pakistan.**

**May, 2023**



***In the name of Allah, the Entirely Merciful, the Especially Merciful***

**Department of Computer and Information Sciences,**

**Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad 45650, Pakistan**

**Declaration of Originality**

I hereby declare that the work contained in this thesis and the intellectual content of this thesis are the product of my own work. This thesis has not been previously published in any form nor does it contain any verbatim of the published resources which could be treated as an infringement of the international copyright law.

I also declare that I do understand the terms ‘copyright’ and ‘plagiarism,’ and that in case of any copyright violation or plagiarism found in this work, I will be held fully responsible for the consequences of any such violation.

**Certificate of Approval**

***This is to certify that the work contained in this thesis entitled***

**“Development of a Web Application, Job Recruitment**

**Portal”**

***was carried out by***

**Mohammad Ahsan Junaid**

**Abdul Rehman**

***Under our supervision and that in our opinion, it is fully adequate, in scope and quality, for the degree of BS Computer and Information Sciences from Pakistan Institute of Engineering and Applied Sciences (PIEAS).***

***Approved by:***

**Signature: Supervisor: *Dr. Anila Usman***

**Signature: Co-Supervisor: *Sir Irfan Hameed***

***Verified by:***

**Signature: Head, DCIS: Dr. Javaid Khurshid**

**Stamp:**

**This project is dedicated to our beloved family, teachers and friends, who offered us unconditional love and support throughout the course of this project.**

**Acknowledgement**

First and foremost, I would like to thank Allah, our sole creator, helper, guider, endower of everything that we have in our lives and forgiver for all the sins that we are guilty of committing. If it wasn’t for His enormous bounty, we would have never achieved anything in our lives and be lost.

I express my sheer appreciation to my supervisor Dr. Anila Usman. She has always been keen on helping and has provided me the guidance on every aspect concerning my project. Even with her grueling schedule she has always taken out time to clear away our doubts or solved any problem which impeded my work.

I also express my sheer appreciation to my co-supervisor, Sir Irfan Hameed. He has always been there for me whenever I needed him, and he has provided me with the necessary guidance on how to proceed with my project. He has helped me immensely in understanding the fundamentals of web development.

Lastly but most importantly I would express my eternal gratitude towards my parents who have always been our backbone throughout the years and have never let me down no matter what.

**Mohammad Ahsan Junaid**

**Abdul Rehman Asif**

**PIEAS, Nilore, Islamabad**

**Table of Contents**

Contents

[Abstract 1](#_Toc135824830)

[Chapter 1: Introduction 2](#_Toc135824831)

[1.1. Problem Definition 2](#_Toc135824832)

[1.2. Proposed System 2](#_Toc135824833)

[System Functions 3](#_Toc135824834)

[Chapter 2: Basic Concepts and Tech Review 4](#_Toc135824835)

[2.1. Information System 4](#_Toc135824836)

[2.2. Programming Languages and Frameworks 4](#_Toc135824837)

[2.2.1. HTML 4](#_Toc135824838)

[2.2.2. CSS 5](#_Toc135824839)

[3.1. Purpose 10](#_Toc135824840)

[3.2 Intended Audience 10](#_Toc135824841)

[3.3 Intended Use 10](#_Toc135824842)

[3.4 Scope 11](#_Toc135824843)

[3.5. Definitions and Acronyms 12](#_Toc135824844)

[3.6. Overall Description 13](#_Toc135824845)

[3.6.1 User Needs 13](#_Toc135824846)

[3.6.2. Assumptions: 13](#_Toc135824847)

[3.7. Functional Requirements 13](#_Toc135824848)

[3.7.1. User Management 13](#_Toc135824849)

[3.7.2. Departments in the System 13](#_Toc135824850)

[3.7.3. Programs in the System 13](#_Toc135824851)

[3.7.4. Store Data of Employees 13](#_Toc135824852)

[3.7.5. Employee Designations 14](#_Toc135824853)

[3.7.6. Store Data of Courses 14](#_Toc135824854)

[3.7.7. Calculating Workload of Faculty Member for Each Semester 14](#_Toc135824855)

[3.7.8. Calculating Payment an Employee in Each Semester. 15](#_Toc135824856)

[3.7.9. Report Generation 15](#_Toc135824857)

[3.7.10. Full/Partial Course 16](#_Toc135824858)

[3.7.11. Separate Page for Configurating Workload Parameters, Employee Payrates and Course Programs 16](#_Toc135824859)

[3.7.12. Printing Appropriate Reports 16](#_Toc135824860)

[3.7.13. Filtering Information 16](#_Toc135824861)

[3.8. Non-Functional Requirements 17](#_Toc135824862)

[3.8.1. Default Departments in the System 17](#_Toc135824863)

[3.8.2. Default Programs in the System 17](#_Toc135824864)

[3.8.3. Default Designations in the System 17](#_Toc135824865)

[3.8.4. User Logs Access and Account Creation Privileges 17](#_Toc135824866)

[3.8.5. Reliability 17](#_Toc135824867)

[3.8.6. Availability 17](#_Toc135824868)

[3.8.7. Maintainability 17](#_Toc135824869)

[3.8.8. Usability 17](#_Toc135824870)

[Chapter 4: Implementation 18](#_Toc135824871)

[4.1. Technology Stack 18](#_Toc135824872)

[4.2. Architectural Designs of the System 18](#_Toc135824873)

[4.2.1. Entities 18](#_Toc135824874)

[4.2.2. Entity Relationship Diagram 20](#_Toc135824875)

[4.2.3. User Interface Structure 21](#_Toc135824876)

[4.3. Challenges 22](#_Toc135824877)

[4.3.1. How to Calculate Workload? 22](#_Toc135824878)

[4.3.2. How to Handle Workload Weights? 22](#_Toc135824879)

[4.3.3. How to Handle Full/Half Courses? 22](#_Toc135824880)

[4.3.4. How to Calculate Payment Due? 22](#_Toc135824881)

[4.4. System Features 23](#_Toc135824882)

[4.4.1. Home Page 23](#_Toc135824883)

[4.4.2. Employee Page 23](#_Toc135824884)

[4.4.3. Add Employee Page 24](#_Toc135824885)

[4.4.4. Edit Employee Page 25](#_Toc135824886)

[4.4.5. Course Page 25](#_Toc135824887)

[4.4.6. Add Course Page 26](#_Toc135824888)

[4.4.7. Edit Course Page 27](#_Toc135824889)

[4.4.8. Workload Page 27](#_Toc135824890)

[4.4.9. Add Workload Page 30](#_Toc135824891)

[4.4.10. Edit Workload Page 31](#_Toc135824892)

[4.4.11. Payment Page 31](#_Toc135824893)

[4.4.12. Report Page 32](#_Toc135824894)

[4.4.13. Configuration Page 35](#_Toc135824895)

[References 40](#_Toc135824896)

[Appendix B – React JS Installation and Setup 43](#_Toc135824897)

[Appendix C – Libraries Used in Project 45](#_Toc135824898)

[● Bootstrap 45](#_Toc135824899)

**Table of Figures**

[Figure 2.1. HTML Page 5](#_bookmark11)

[Figure 2.2. CSS Code 6](#_bookmark13)

[Figure 2.3. Salient Features of React JS 8](#_bookmark18)

[Figure 2.4. Collections in MongoDB 9](#_bookmark20)

[Figure 3.1. Intended Use of the System 10](#_bookmark25)

[Figure 4.1. Employee Entity 19](#_bookmark60)

[Figure 4.2. Course Entity 20](#_bookmark61)

[Figure 4.3. Workload Entity 20](#_bookmark62)

[Figure 4.4. Payment Entity 21](#_bookmark63)

[Figure 4.5. ER Diagram of the System 22](#_bookmark65)

[Figure 4.6. User Interface Structure 23](#_bookmark67)

[Figure 4.7. Home Page 25](#_bookmark75)

[Figure 4.8. Employee Page 26](#_bookmark77)

[Figure 4.9. Add Employee Page 26](#_bookmark79)

[Figure 4.10. Edit Employee Page 27](#_bookmark81)

[Figure 4.11. Course Page 28](#_bookmark84)

[Figure 4.12. Add Course Page 28](#_bookmark85)

[Figure 4.13. Edit Course Page 29](#_bookmark88)

[Figure 4.14(a) Workload Page 30](#_bookmark89)

[Figure 4.14(b) Workload Page 31](#_bookmark90)

[Figure 4.15. Add Workload Page 32](#_bookmark92)

[Figure 4.16. Edit Workload Page 33](#_bookmark95)

[Figure 4.17. Payment Page 34](#_bookmark97)

[Figure 4.18(a) Report Page 35](#_bookmark98)

[Figure 4. 18(b) Report Page 35](#_bookmark99)

[Figure 4.18(c) Report Page 36](#_bookmark100)

[Figure 4.18(d) Report Page 36](#_bookmark101)

[Figure 4.19(a) Configuration Page 37](#_bookmark103)

[Figure 4.19(b) Configuration Page 38](#_bookmark104)

[Figure 4.19(c) Configuration Page 39](#_bookmark105)

# Abstract

The Job Recruitment Portal is a comprehensive web-based platform designed to help employers and applicants streamline the job recruitment process. The portal aims to connect job seekers and employers through an efficient and user-friendly interface, facilitating seamless job search, application, and selection procedures. The administration, employers, and applicants are the three main actors in the system. As the super user, the admin has the authority to manage the database, monitor system statistics, and perform other administrative tasks. Employers can post job openings, review applicant profiles, select or reject applicants, and download resumes as well as contact applicants through the in-built mailbox. Applicants, on the other hand, can search available job listings, apply for relevant positions, and showcase their qualifications through resume submissions. The Job Recruitment Portal aims to improve the efficiency and effectiveness of the job recruitment process for both employers and applicants by providing a centralized platform for job posting and application management. To ensure a robust and password protected secure system, the project employs a number of technologies and frameworks. We investigate the design, development, and evaluation of the Job Recruitment Portal, as well as its performance and potential future enhancements, in this thesis. The outcomes demonstrate the system's effectiveness in facilitating seamless job recruitment procedures and improving the overall experience for both employers and applicants.

# Chapter 1: Introduction

The Job Recruitment Portal aims to revolutionize the job recruitment process by providing employers and applicants with a user-friendly and efficient platform. Traditional methods of recruitment frequently result in delays, inefficiencies, and missed opportunities. This project addresses these issues by providing a centralized system with three primary actors: administrator, employers, and applicants. The database and system statistics are managed by the administrator, while employers can post jobs, review applicants, and download resumes. Applicants can look for jobs and apply for them. The portal makes use of technologies and frameworks to ensure a stable system. The focus of evaluation is on performance, usability, and user satisfaction. The findings contribute to the field of job recruitment and provide guidance for future improvements. The goal is to improve the overall experience for employers and applicants by streamlining the recruitment process, saving time and effort.

## Problem Definition

Traditional recruitment methods frequently face a number of challenges that impede the process's efficiency and effectiveness.

* Handling a large volume of job applications manually is time-consuming and labor-intensive, resulting in delays and potential oversight of qualified candidates.
* Job seekers struggle to find relevant opportunities among numerous sources, resulting in potential positions being missed.
* The lack of standardized formats for resumes and job postings makes objective candidate evaluation and decision-making difficult.
* Without a centralized system, employers struggle to effectively track and manage applications.
* Limited Reach and Narrow Candidate Pool: Conventional techniques, such as newspaper ads or physical job postings, may only reach a narrow pool of individuals. Due to this, there may be a lack of diversity and skilled applicants who are not actively looking for work through conventional routes may go unnoticed.
* High costs: Conventional hiring practices can be pricey, including participating in job fairs and newspaper advertising. Advertising, travel, and other associated costs may require organizations to set aside large sums. This can be difficult, especially for startups and small businesses with scarce resources.
* The management of applications, paperwork, and communication can be inefficient and prone to mistakes when done manually. Traditional techniques frequently lack automated components.

## Proposed System

The solution for this is a modern “Job Recruitment Portal”, As the world moves on to a digital era, searching for Jobs or finding the right Employee for one, like all other aspects of daily life, are being shifted to be online, saving time, human resources, accessible and providing ease of use. We intended to create a platform that will automate the process where organizations can put up job vacancies, where applicants can go through them and apply in the right job.

Where employers would have the ease of checking through CV’s and selecting the suitable candidates etc. And applicants will find a platform that will allow them to apply in multiple jobs according to their qualifications easily as well as get side-by-side updates.

It is web-based platform, with a good UI/UX having a Landing page, Job Vacancies Page from where one can apply into preferred job after registering with the system as an applicant where email validation shall be done, then provide data and upload their CV and apply for jobs they want, they can also search by the search bar and apply search filters for their needs like locations, experience requirements etc. Employers would have access to a job posting page where they set the requirements and the details of the job like title, salary, details etc. This system has a database for all records of jobs i.e. Job Titles (with details), Applications (CV attached, Data provided) as well as details of users such as employers, and most importantly the applicants like their Names, CNIC, Addresses, Qualifications, and so on.

To secure all this data and avoid mishaps, security measures will be taken starting off with proper Authorization being implemented like access-controls over user types, where an applicant can only view, update his/her own data rather than having access of others, the employer only accessing data of applicants who have applied to his posted job, Admin having access of everything including the database etc.

Applicants/Candidates can apply in one or more jobs with easy without entering same data (name, CNIC, Addresses etc.) again for each job they apply. On the employers end they can check all the applications to a certain job, contact applicants regarding concerns through in-built communication channel and download CV’s and data of applicants for further processing if needed.

## System Functions

The following are key system functions:

* **User Registration and Login Authentication:**

Employers and Applicants can register and create accounts on the portal, that are password protected and encrypted in database, hence even admin cannot see them.

User login authentication ensures secure system access and protects user data.

* **Job Posting and Management:**

Employers can create and post job openings, describing the job's details and requirements.

As needed, job listings can be edited, updated, or removed.

* **Filtering and Job Search:**

Applicants can use specific criteria to search for available jobs. Filtering options allow applicants to refine their search results and find relevant job opportunities.

* **Management of Applicant Profiles:**

Employers can review applicant profiles, including resumes, can establish contact if they have applied to any job posting by that employer.

* **Candidate Selection:**

Based on their suitability for the job, employers can mark applicants as selected / rejected.

Selected candidates can be contacted for interviews, assessments, or other stages of the selection process through the in-built mailbox.

* **Download Resume/CV:**

Employers can download applicants' resumes or CVs for offline review or further evaluation.

* **Management and the Admin Dashboard:**

The system administrator has access to a comprehensive dashboard for system management.

Admin functions include approval of employer account, managing users, databases, and generating system statistics. Admin reserves rights to delete any posted job, or employer account as well from here.

* **In Built Mailbox:**

The in-built mailbox offers applicants and employers to communicate any misunderstandings or important notices and details.

* **Change Password / Deactivate Accounts:**

Employers and applicants can both use this option to modify their passwords inside the portal. To ensure account security, users can reset their passwords and access their account settings. Users are also able to deactivate their accounts within the portal.

# Chapter 2: Basic Concepts and Tech Review

Chapter 2 contains many basic concepts, technologies and frameworks that are significant in the conduct and development of the project titled “*Information System for Managing Visiting Faculty / Teaching Assistants”.*

## Information System

Information systems are large, linked collections of information, data and processes. They are employed in virtually every element of human life, business, and industry.

The phrase "information system" in technology refers to any tool or information system that facilitates the gathering and utilization of data. Information systems can be utilized to give aid inside a company or for personal benefit.

A good information system enables the user to quickly access, comprehend, and react to information. Users get access to the most up-to-date information whenever they need it to complete a task.

## Programming Languages and Frameworks

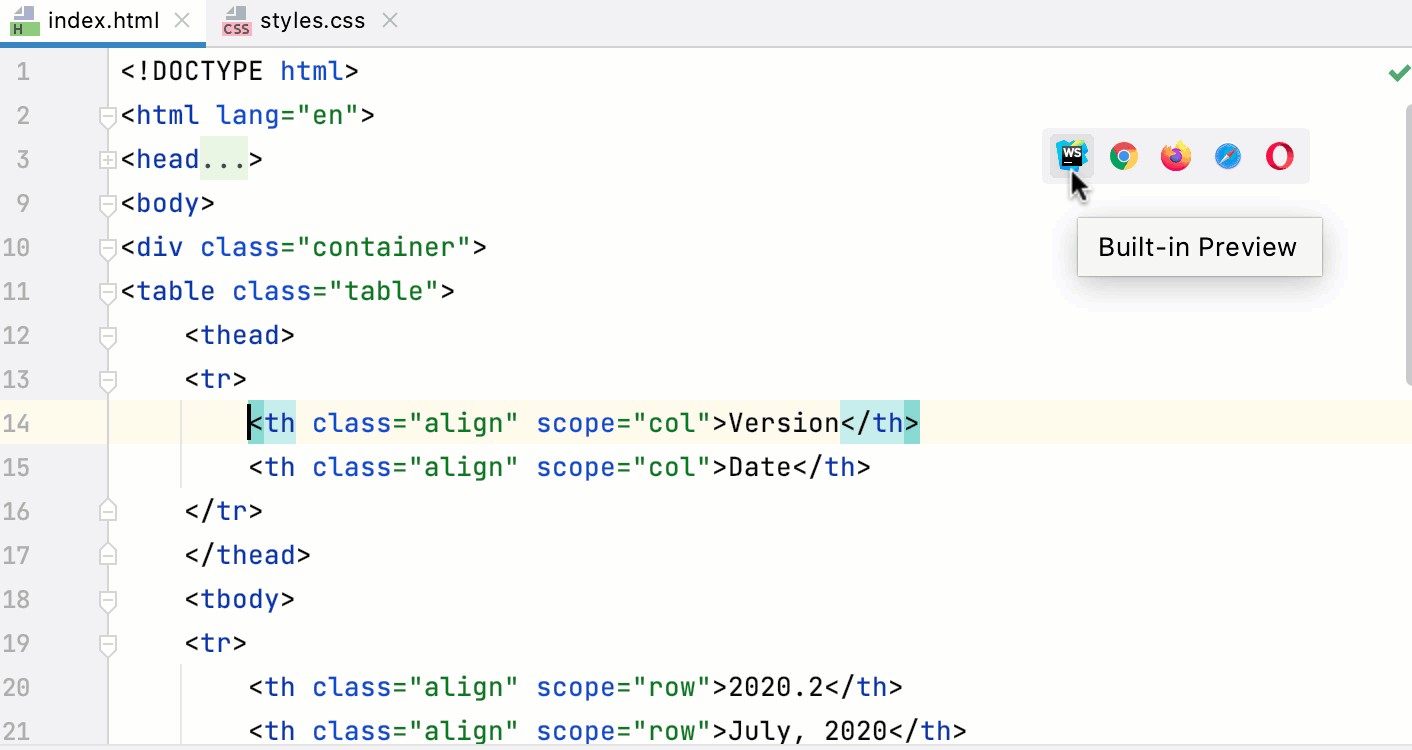
In this section, different technologies and tools that can be used to develop this information system are discussed.

### HTML

If you decide to create a web-fronted app for mobile devices, HTML is the best option. Given that it is not a programming language, java script commonly uses it. It enables you to logically arrange all of the content on your pages. It works with any browser and makes your pages look their best across all platforms.

The foundation of HTML pages are HTML elements. Images and other objects, like interactive forms, may be embedded within the rendered pages. By indicating structural semantics for text elements like headings, paragraphs, lists, links, quotations, and other objects, HTML offers a way to generate structured texts. Tags, which are written in angle brackets, are used to distinguish HTML elements. Input and image tags, for example, add content directly to the page. Other tags, like <p> describe the text of the document. They may also contain other tags as sub-elements. Browsers use the HTML tags to interpret the page's content rather than displaying them.

The figure 2.1 shows an example of HTML page;



### CSS

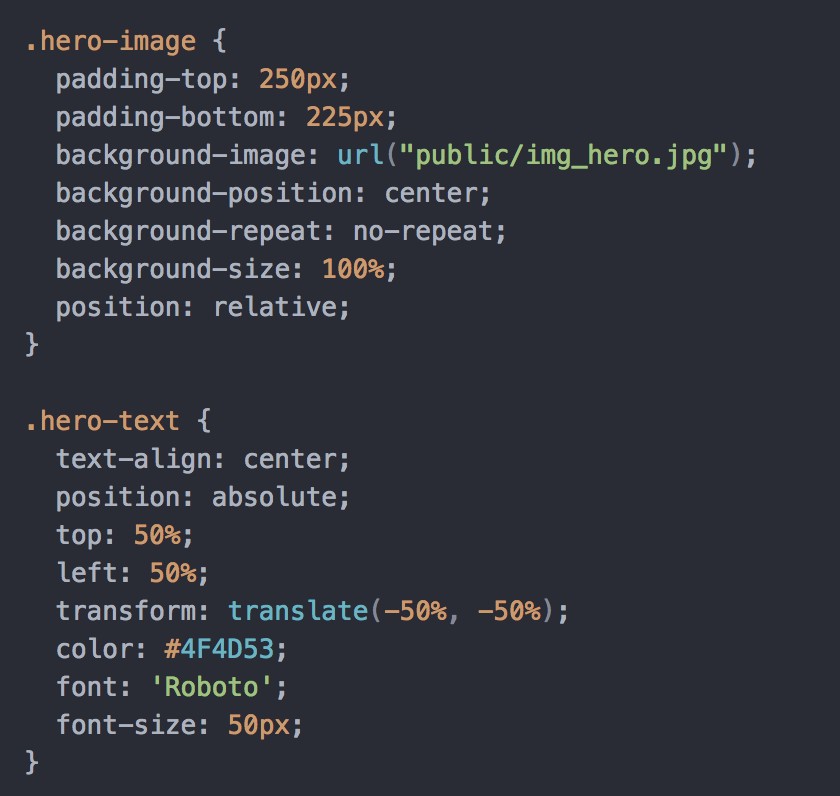
*Figure 2.1. HTML Page [1]*

A stylesheet language called Cascading Style Sheets (CSS) is used to describe how a document written in HTML is presented. CSS specifies how items should be shown in various media, including speech, paper, screens, and other media.

The style and feel of a web page are handled by CSS. The color of the text, the font style, the spacing between paragraphs, the size and arrangement of columns, the background pictures or colors used, layout designs, differences in display for various devices and screen sizes, and a variety of other effects can all be controlled using CSS.

Although CSS is simple to grasp and learn, it offers strong control over how an HTML document is presented. CSS is most frequently used in conjunction with HTML as markup languages.

The figure 2.2 shows a CSS code used to style an HTML page;



* + 1. **Tailwind CSS**

*Figure 2.2. CSS Code [2]*

The quickest and simplest method to create webpages is with Tailwind CSS. Tailwind CSS is a utility-first CSS framework for quickly creating unique user experiences. It is a low-level CSS framework that is extremely adaptable and provides all the building blocks required to create custom designs without requiring you to struggle to overcome obnoxious opinionated styles.

The best part about tailwind is that it doesn't enforce design guidelines or dictate how your website should look; instead, you just combine little parts to create a one-of-a-kind user interface. Simply put, Tailwind takes a "raw" CSS file, processes it through a configuration file, and outputs the results.

* + 1. **Bootstrap**

Bootstrap is a powerful front-end framework for building contemporary web pages and online apps. It's open-source and free to use, but it includes plenty of HTML and CSS templates for UI interface components like buttons and forms. JavaScript extensions are supported by Bootstrap as well.

The frontend of our information system is built using bootstrap 5 which is the latest version. The grid system of bootstrap really helped in creating responsive web pages that respond to the size of the screen.

* + 1. **Material UI**

To develop a user interface in our React applications, we can simply import and use several components from the Material-UI package. As a result, the developers can save a lot of time by not having to write everything from start.

Google's guidelines for creating user interfaces served as a major source of inspiration for Material-UI widgets. Therefore, creating aesthetically appealing applications is simple for developers.

* + 1. **JavaScript**

JavaScript is a dynamic language that supports object-oriented, declarative (like functional programming), and imperative programming paradigms.

The ECMAScript Language Specification and the ECMAScript Internationalization API specification are the standards for JavaScript.

The wide library of frameworks available in JavaScript enables programmers to create mobile and online applications quickly. Developers use frameworks, which are collections of pre- written JavaScript code, for common functionalities.

The most widely used frameworks for JavaScript applications are

* + - 1. React is a group of JavaScript tools designed for creating web application user interfaces.
      2. React Native is a collection that enables JavaScript developers to create mobile apps.
      3. Node.js is a set of tools that permits two-way data exchange with servers.

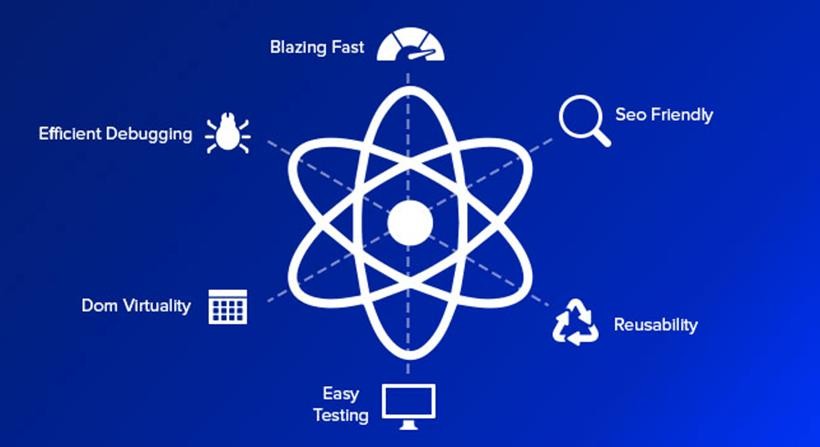
Some of the most potent social networking systems in use today are powered on Facebook's React and React Native frameworks, which were created by the company. Uber, Pinterest, Netflix, Instagram, Amazon, Twitter, Udemy, and many other websites use React.

* + 1. **React JS**

One of the top frameworks for building web applications is react.js. It helps programmers to design interactive components. Features of React.js also increase general productivity.

React JS is a JavaScript library for creating reusable UI components that is fast, and versatile. It is an open-source, component-based front-end library that is exclusively responsible for the application's view layer. It was created and maintained by Facebook and was later utilized in its products such as WhatsApp and Instagram.

The primary goal of ReactJS is to create User Interfaces (UI) that increase app performance. It makes use of virtual DOM (JavaScript object), which increases the app's speed. The virtual DOM in JavaScript is quicker than the conventional DOM. ReactJS may be used on the client and server sides, as well as in conjunction with other frameworks. It employs component and data patterns that increase readability and aid in the maintenance of bigger apps. The figure 2.3 elaborates salient feature of React JS.



*Figure 2.3. Salient Features of React JS [3]*

In the developed project, React JS has been used to create and maintain different UI components using “Axios” and “React Dom” and other libraries like “Use State” to maintain the state of various objects and components.

* + 1. **MongoDB**

MongoDB is a document-oriented NoSQL database that is used for large-scale data storage. In contrast to conventional relational databases, MongoDB uses collections and documents rather than tables and rows. Key-value pairs, the fundamental building block of data in MongoDB, are composed of documents. Collections are sets of documents and functions that are the equivalent of relational database tables. MongoDB is a database that first appeared in the mid- 2000s.

Collections, which serve as the equivalent of relational database tables, are groups of documents. Data in a collection cannot be distributed across many databases, despite the fact that collections can contain any kind of data.

Users can query and update data, perform administrative tasks, and communicate with MongoDB.

An example of key value pairs is shown in figure 2.4.



*Figure 2.4. Collections in MongoDB [4]*

**Chapter 3: Software Requirements Specifications**

## 3.1. Purpose

The purpose of the project is to develop an online information system that provides all the necessary information or details of the activities that an employee is currently involved in and manages and define workload for regular faculty, visiting faculty and teaching assistants / lab engineers based on designation, labs, thesis supervision, managerial responsibilities, other contributions etc. and helps in managing and automating payments of visiting faculty and TA/lab engineers.

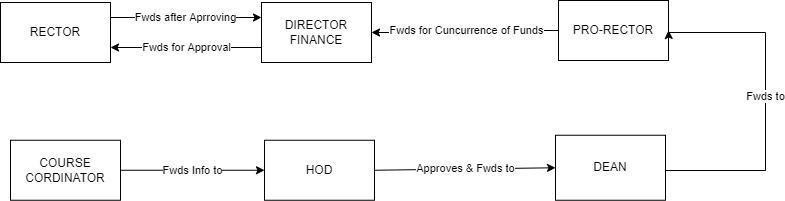
## Intended Audience

The intended audience of the system is following:

* + 1. Course Coordinator (CC)
    2. Head of Department (HoD)
    3. Dean
    4. Pro-Rector
    5. Director Finance
    6. Rector

## Intended Use

The figure 3.1 elaborates how these users will be using the information that the system will generate.



*Figure 3.1. Intended Use of the System*

## Scope

In terms of the information management system, this project has a very broad scope. Overall, it will entail learning and experimenting with

* + 1. Web design, setup and development.
    2. Database creation and maintenance.
    3. Design and setup of the server.

Internally, there are two modules to this project's development.

1. Workload
2. Payment
   * 1. **Workload**

Workload module aims to provide all the necessary information or details of the activities that an employee is currently involved in. Secondly, it will also manage and define workload for regular faculty, visiting faculty and teaching assistants / lab engineers.

* + 1. **Payment**

Payment method involves financial impact which is basically the payment of an employee according to the expected classes and payment due which basically is the payment of an employee according to the actual classes that an employee has taken at the end of semester.

## Definitions and Acronyms

Following are the acronyms and their definitions that are used in this final report of the project.

|  |  |  |
| --- | --- | --- |
| **#** | **Acronym** | **Definition** |
| 1 | CC | A course coordinator's major responsibility is to maintain quality and  consistency of instruction in multiple-department courses. |
| 2 | HoD | A faculty member who has been granted the responsibility of leading  a particular academic department. |
| 3 | Dean | Deans have academic, managerial, and fiscal responsibilities for a  university and also verify the correctness of instruction. |
| 4 | Pro-Rector | The deputy Rector in a university. |
| 5 | Director  Finance | A director of finance administers the financial operations and  financial planning of a company. |
| 6 | Rector | The Rector has a supervisory role and also represents the university. |
| 7. | Workload | It is a numerical value that is calculated from different factors  including courses, managerial position etc. |
| 8. | Contact  Hours | Total hours to which an employee is engaged to a particular course. |
| 9. | Financial  Impact | Total expected payment of an employee at the start of the semester. |
| 10. | Payment Due | Total actual payment to be issued to an employee at the end of the  semester. |
| 11. | Course  Contribution | It describes whether an employee is fully or partially engaged in a  course. |

## Overall Description

### User Needs

* + - 1. Central repository of payments history and reports.
      2. Overview of workload burden of all employees; to simplify the assignment of courses.
      3. Automatic calculation of department expenses.

### Assumptions:

Following are the assumptions of the developed project;

* + - 1. System is only be used by elevated users.
      2. Course Coordinator has knowledge about course schedules, teacher’s miscellaneous details such as research projects, total students taught, etc.
      3. Prior knowledge about how the system works.
      4. Two or more courses cannot have the same course code.
      5. Course Coordinator is aware of missed no. of classes for each teacher for finalizing payments.

## Functional Requirements

### User Management

Central User Management will be incorporated with appropriate access/privileges of the system.

### Departments in the System

Both employee and course will require a department's field that must be associated with it. This will play an important role when generating department-wise financial reports.

### Programs in the System

A department can have many programs. There must be an option to add or remove programs dynamically as needed. A separate page for such purposes is preferred.

### Store Data of Employees

The developed information system will take inputs from a user through a form and store all the employees and their relevant information in the database. Moreover, the developed system will also show this data in the form of table and will be capable to update and delete them. Employee attributes are name, department and designation.

### Employee Designations

Each designation will be assigned a pay rate on the basis of which payment for each employee would be made in the payment module. One thing which is to be noted is that the payment module is only for visiting faculty and lab/engineers as there is a system that is already developed for the regular faculty.

### Store Data of Courses

The Information system will take inputs from a user through a form and store all the courses and their relevant information in the database. Moreover, this system will also show this data in the form of table and will be capable to update and delete them. Course attributes are title, department, program, theory and lab credit hours.

### Calculating Workload of Faculty Member for Each Semester

The information system will also take input from a course coordinator and store all the workloads and their relevant information of an employee of a particular semester in the database. Moreover, the system will also show this data in the form of table and is also capable update as well as delete them.

**Factors Contributing to Workload Final Score**

After filling the workload form, system will generate a final score. Based upon these factors, workload will be computed by the system

* + - 1. **Teaching**
         1. Total Contact Hours
         2. Total Number of Students (BS & MS/PhD)
      2. **Research And Development (R&D)**
         1. Project Supervisions (BS & MS/PhD)
         2. Publications (National Journal, National Conf., Int. Journal, Int. Conf.)
         3. Research Project (< 1M, 1-2M, 2-5M, 5-10M, 10M+)
         4. Book (National & International)
         5. Patent (National & International)
         6. Technical Report
         7. Book Chapter
         8. Journal / Conference Reviewer
      3. **Administrative/ Managerial Responsibility**
         1. Rector / Pro-Rector
         2. Dean / Director
         3. Registrar / Controller
         4. Head of Department
         5. Head of Division
         6. Head / In charge of Section
         7. Project Director
         8. Course Coordinator / Project Coordinator
         9. Conference / Short Course Coordinator
         10. Conference / Short Course Team Member
         11. Visit Coordinator
         12. One Day Seminar Organizer
         13. Focal Person

### Calculating Payment an Employee in Each Semester.

The information system will help in estimating payments of visiting faculty and TA/lab engineers by calculating;

* + - 1. **Financial Impact =** [(Total Weeks) \* (Total Contact Hours) \* (Pay Per Hour)]
      2. **Payment Due =** [Total Weeks \* Total Contact Hours \* Pay Per Hour] **±** [Classes] where

1. Total Contact Hours = [Theory + 3 (Lab)] Credit Hours
2. Total Weeks in Semester = 16
3. Pay Per Hour is retrieved according to the employee designation/category

### Report Generation

A report will be generated for each faculty member for each department. This report will categorize information in;

* + - 1. Financial Impact for Visiting Faculty
      2. Financial Impact for TA/ Lab Engineers
      3. Payment Due for Visiting Faculty
      4. Payment Due for TA/ Lab Engineers
      5. Department Wise Summary of Financial Impact
      6. Department Wise Summary of Payment Due

### Full/Partial Course

An employee can teach full/partial course or we can say two employees can also teach one course. Therefore, a method shall be incorporated in the system that determines whether an employee is fully or partially engaged in a course and determines financial impact and payment due accordingly.

### Separate Page for Configurating Workload Parameters, Employee Payrates and Course Programs

A separate section should be added for admin only to;

* + - 1. Add additional programs
      2. Change any Workload Weight Parameter
      3. Change Payrates for Different Employee Designations/Categories

### Printing Appropriate Reports

The information system will also be able to print following

* + - 1. Employee’s Workload Information
      2. All information in Report Section

### Filtering Information

As there can be many entries in a table. Therefore, it becomes hard to find relevant data from big database. Therefore, to tackle this, filter dropdowns must be added so that information relevant to that particular filter can be fetched only.

## Non-Functional Requirements

### Default Departments in the System

### Default Programs in the System

### Default Designations in the System

### User Logs Access and Account Creation Privileges

* + - 1. Only admin is allowed to create accounts and register users on this portal.
      2. Only the admin can see user logs.
      3. Only registered persons can use this system.
      4. All registered users will be given access according to their roles in the system.

### Reliability

The information system will be very reliable after testing it thoroughly at every stage. The system will not crash frequently because there are only a few users and a limited amount of data is being handled through this information system.

### Availability

The product will be web based which means it will be easily accessible from anywhere at any time 24/7.

### Maintainability

The software will be created in such a way that it is easily maintainable and modifiable. So, if there is any need to extend or change the behavior of different components it can be easily done.

### Usability

The user interface is simple yet comprehensive which can be easily understood by every user.

# Chapter 4: Implementation

## Technology Stack

The technologies and frameworks that are used to develop system are;

* + 1. Bootstrap
    2. ReactJS
    3. MongoDB

## Architectural Designs of the System

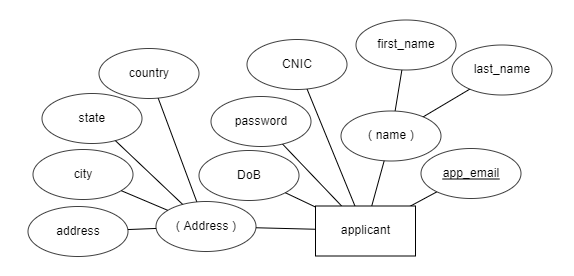
### Entities

A concept or thing within a system, such as a person/role, object, concept, or event, is referred to as an entity.

Following are the entities of the system;

**Applicant**

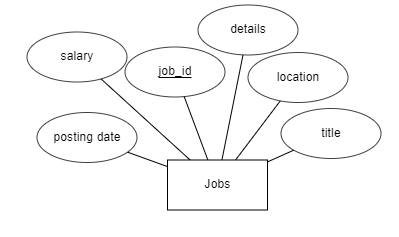
applicant entity diagram , shown in figure, .



*Figure 4.1. Applicant Entity*

**Job Entity**

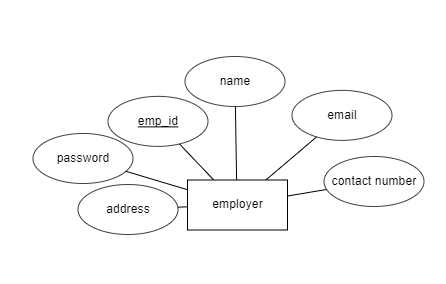
Job entity, shown in figure .



*Figure. Job Entity*

**Employer:**

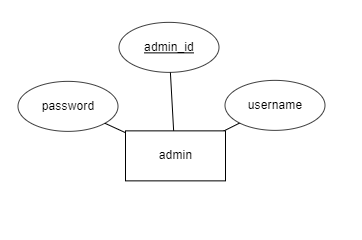
Employer entity, shown in figure



*Figure 4.3. Employer Entity*

**Admin**

Admin entity, shown in figure.

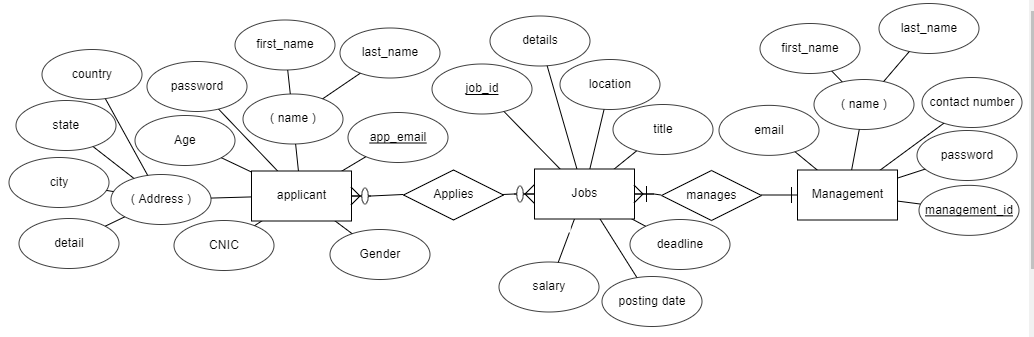


*Figure Admin Entity*

### Entity Relationship Diagram

The Entity Relationship Diagram (ER Diagram) depicts the relationships between entity sets stored in a database. To put it another way, ER diagrams aid in the description of database logical structure. Entities, attributes, and relationships are the three fundamental concepts that underpin ER diagrams.

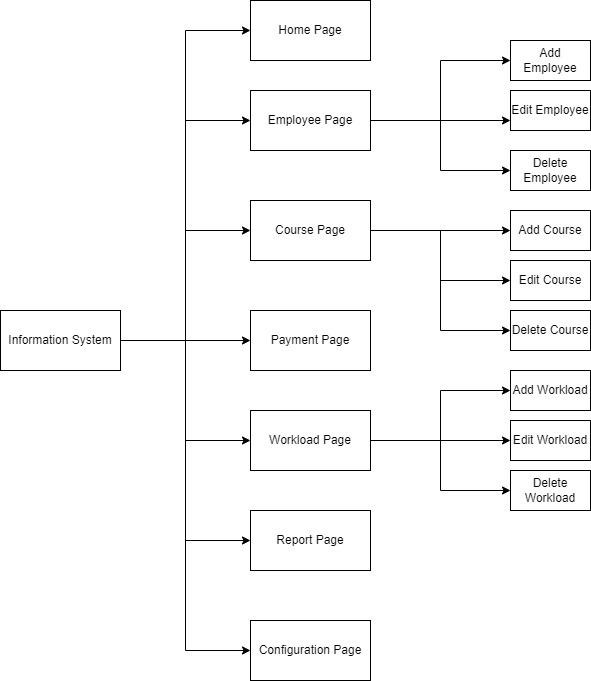
An ER diagram and a flowchart appear similar at first glance. The ER Diagram, on the other hand, is a distinct model due to the numerous specific symbols and their meanings. The ER Diagram depicts the entity framework architecture.



*ER Diagram of the System*

### User Interface Structure

The user interface structure of the system is shown in the figure 4.6;



*Figure 4.6. User Interface Structure*

## Challenges

### How to Calculate Workload?

Workload calculation was one of the challenging tasks as there was no previous system in the institute that calculates workloads of its employees. Moreover, there are several factors that can contribute in workload so how to tackle them and bring into an automated system was also a big challenge for me. For this, a formula was derived to calculate workload’s final score.

Final Score = [Factor 1\*wFactor 1] + [Factor 2\*wFactor 2] + [Factor 3\*wFactor 3] & so on... where

* + - 1. Factor is the value that user types in the workload form.
      2. wFactor is the weight of a particular factor.

### How to Handle Workload Weights?

Each factor of the workload contains its weight. So, if the weights of factors are set static, it becomes a hectic task to change the weight of a factor repeatedly because there may come a need in future to alter any of the weights.

In order to tackle this situation, a configuration page is developed where all the weightage parameters can be updated.

### How to Handle Full/Half Courses?

There were cases where more than one employee is engaged in a course. In order to tackle this situation, a new course contribution field was introduced. This field determines whether an employee is fully or partially engaged in a particular course.

### How to Calculate Payment Due?

Payment due is calculated on the basis of number of classes. In order to calculate correct payment due, following steps are followed;

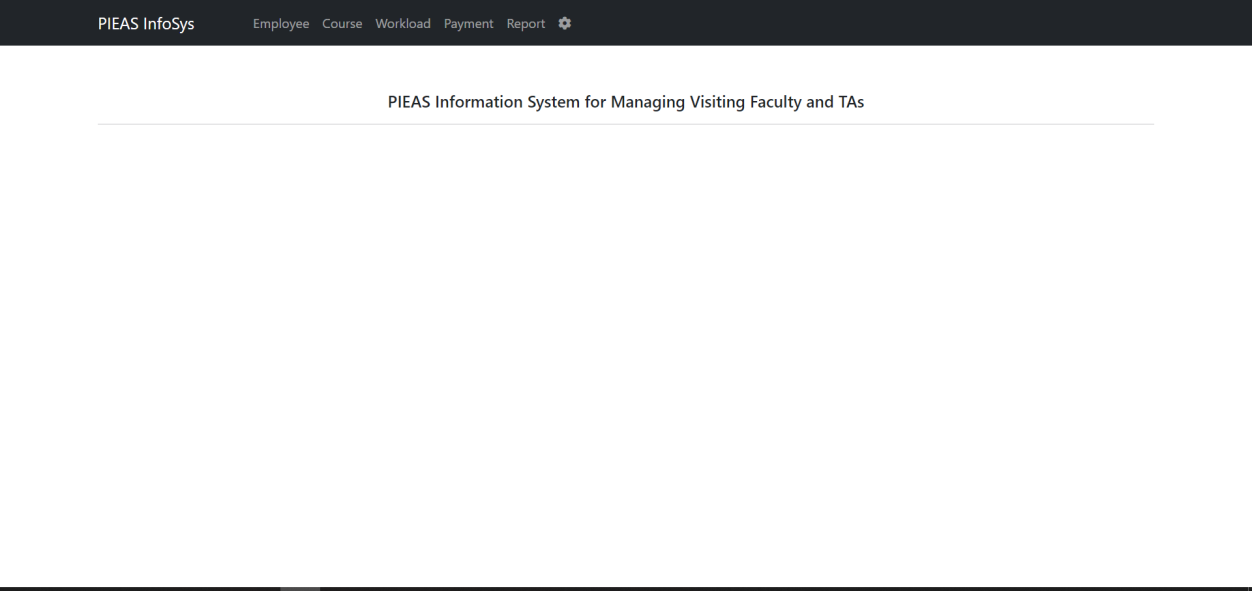
* + - 1. Financial Impact is divided by total number of classes including labs.
      2. A value that is entered by course coordinator will be multiplied with that number.
      3. Payment due is then added to the above value.

## System Features

The features of the information system are following:

### Home Page

Home page is the first page that will be shown to the user. The interface of the login page is shown in figure 4.7.



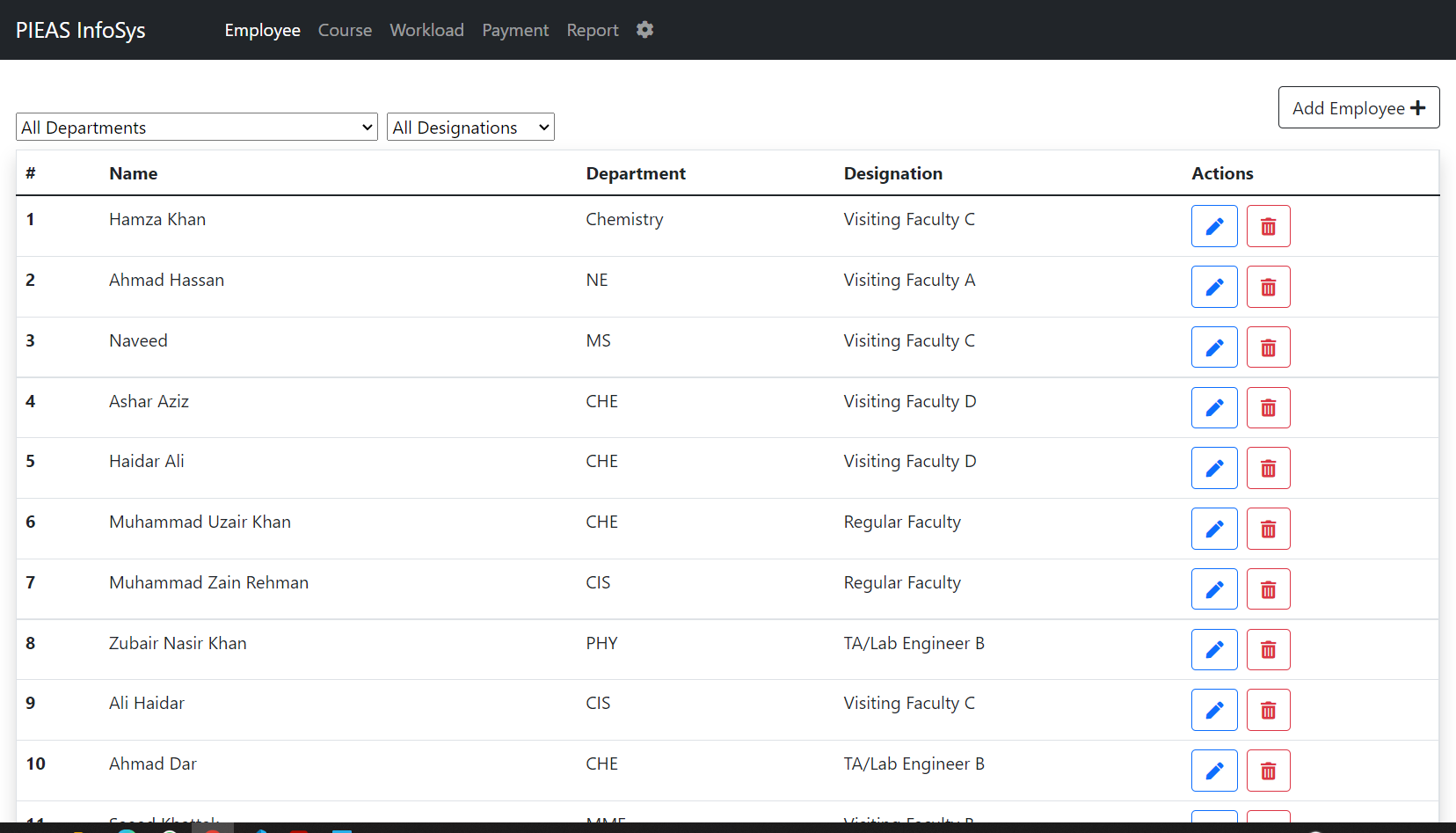
*Figure 4.7. Home Page*

### Employee Page

After navigating the employee page, all the employee’s data is fetched from the database and shown in a table in which we can edit and delete the information of a particular employee as well. User can also filter employees by their departments and designations. The employee has following attributes

* + - 1. Employee Name
      2. Department
      3. Designation

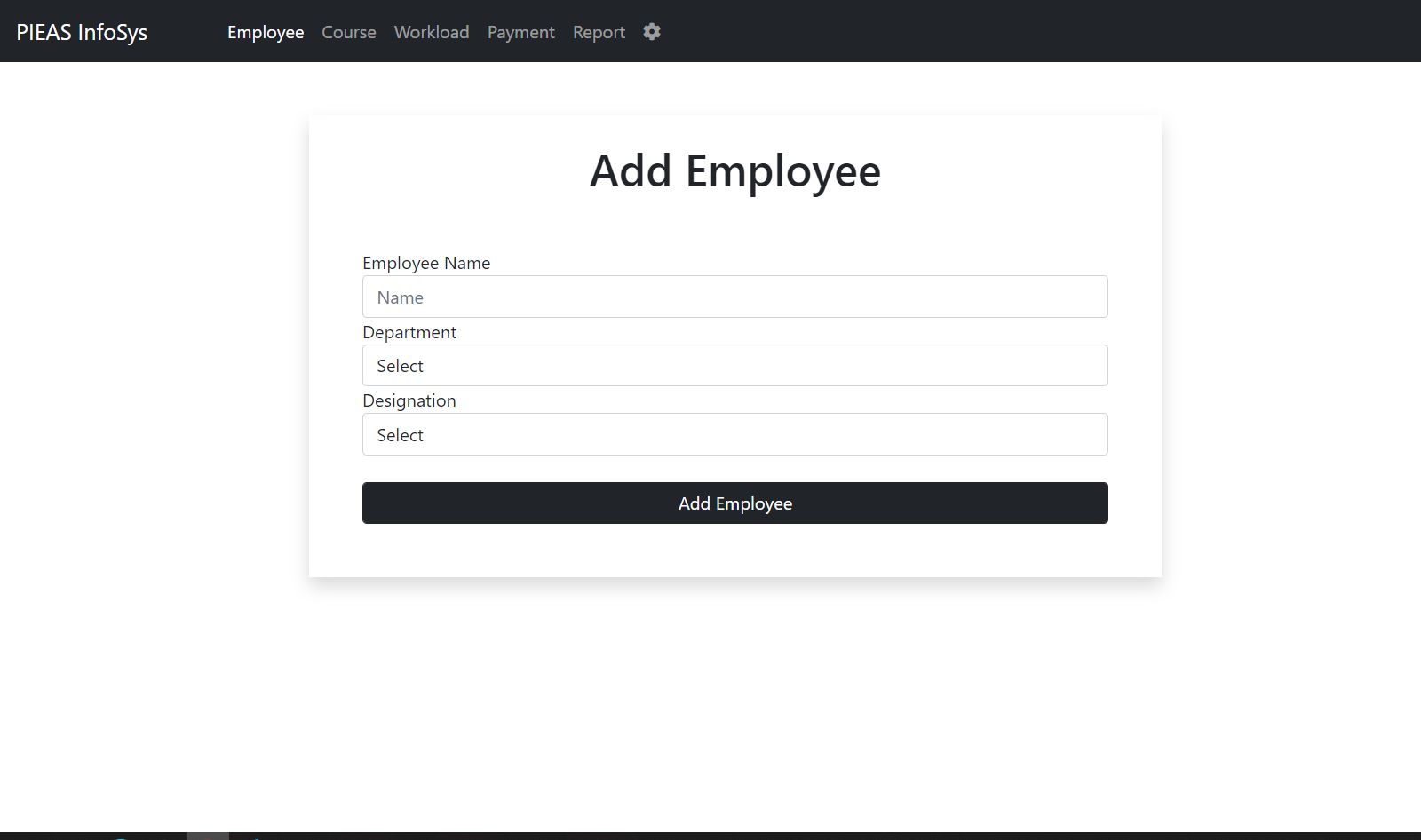
The employee page is shown in figure 4.8.



*Figure 4.8. Employee Page*

### Add Employee Page

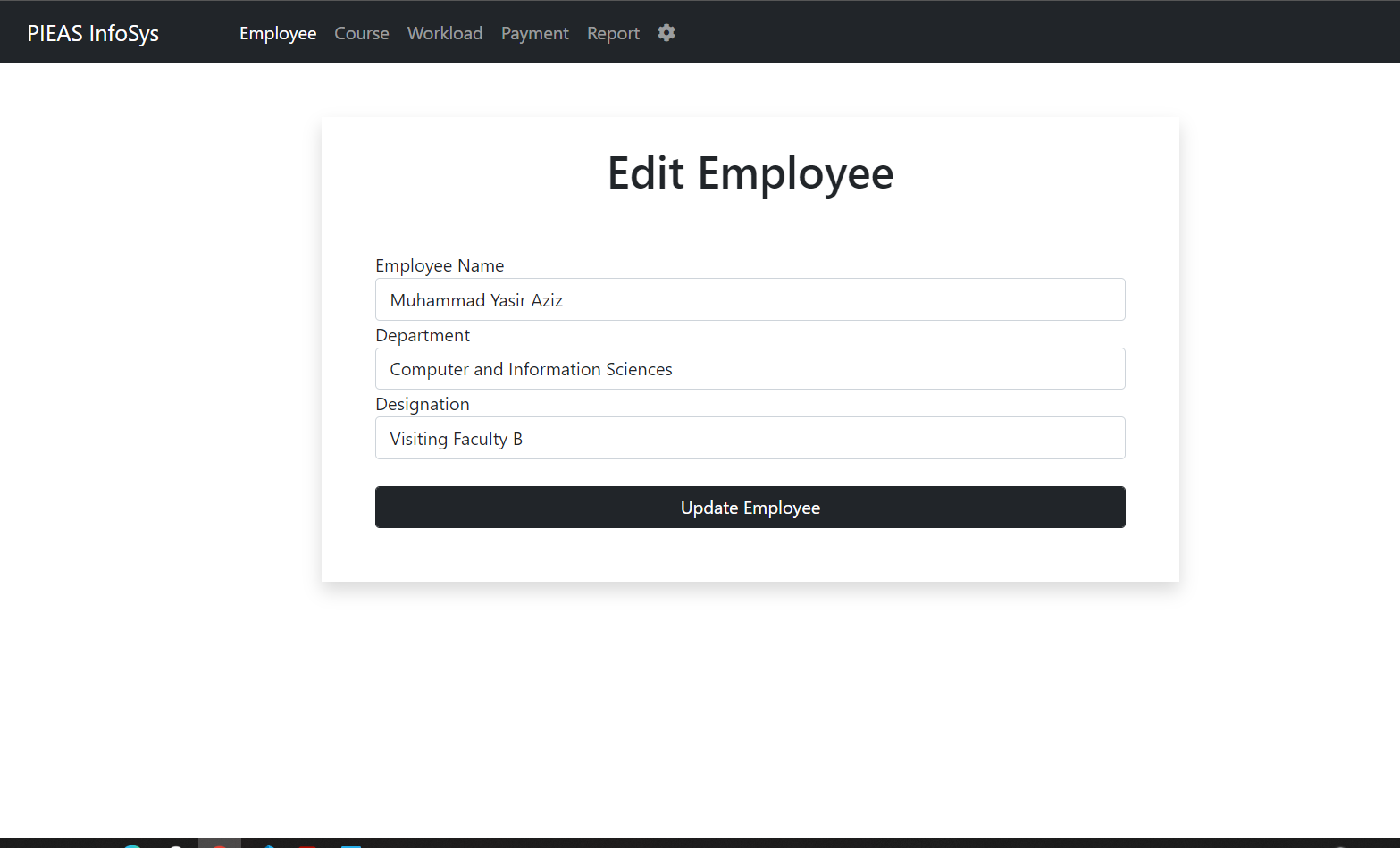
In this page, user can add employee that will be enrolled in the information system. The user interface of this page is shown in the figure 4.9.



*Figure 4.9. Add Employee Page*

### Edit Employee Page

Employee page also contains an edit button that is used to edit a particular employee and that employee entry will be edited in the system. The user interface of this page is shown in figure 4.10.



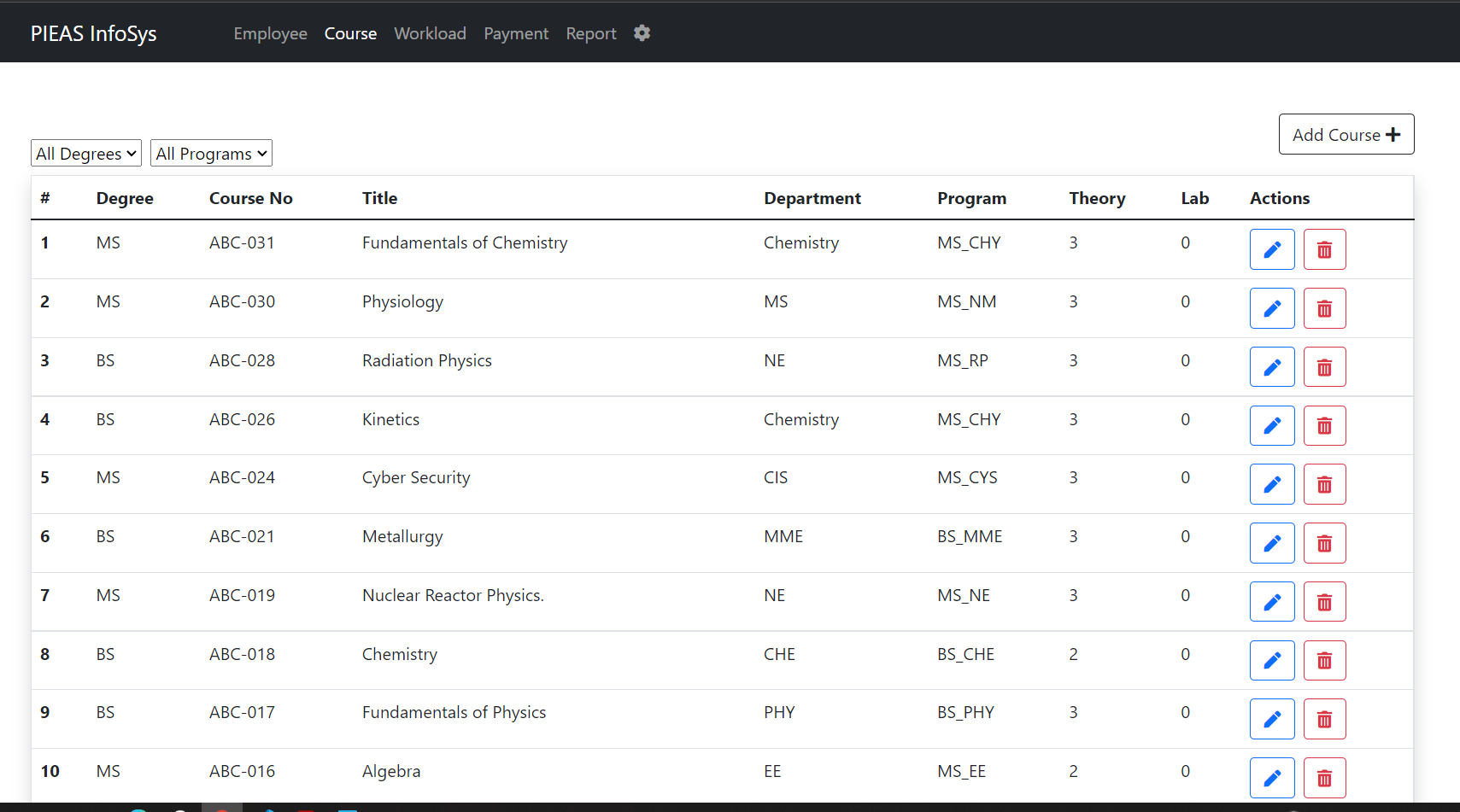
*Figure 4.10. Edit Employee Page*

### Course Page

This page contains all the data regarding courses like its degree, title, department and program etc. that is entered by a user. Users can also filter courses by degree and program for his/her ease. The course contains the following attributes

* + - 1. Degree
      2. Department
      3. Program
      4. Course Number
      5. Course Title
      6. Theory Credit Hours
      7. Lab Credit Hours

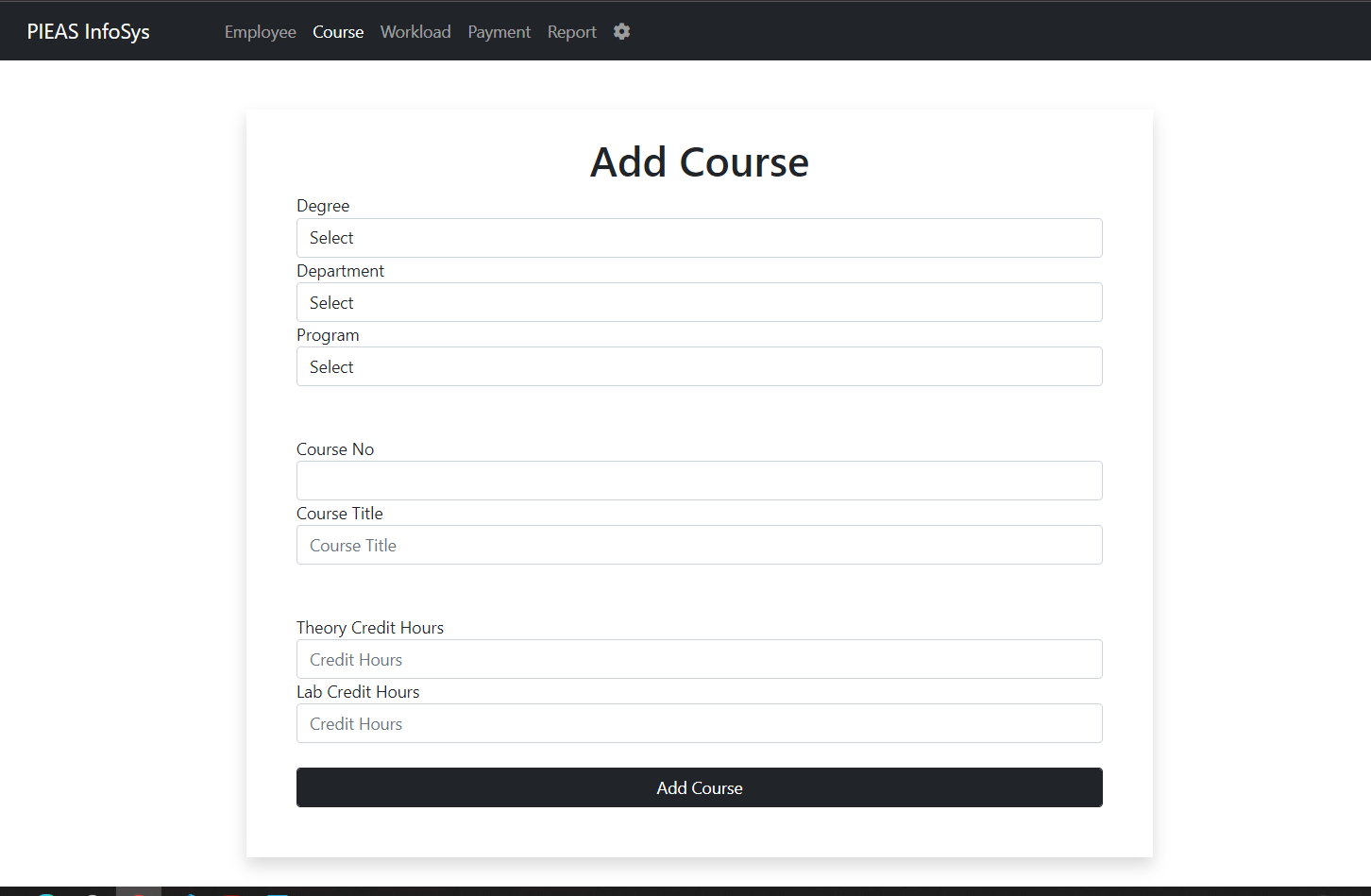
The interface of this page is shown in figure 4.11.



### Add Course Page

*Figure 4.11. Course Page*

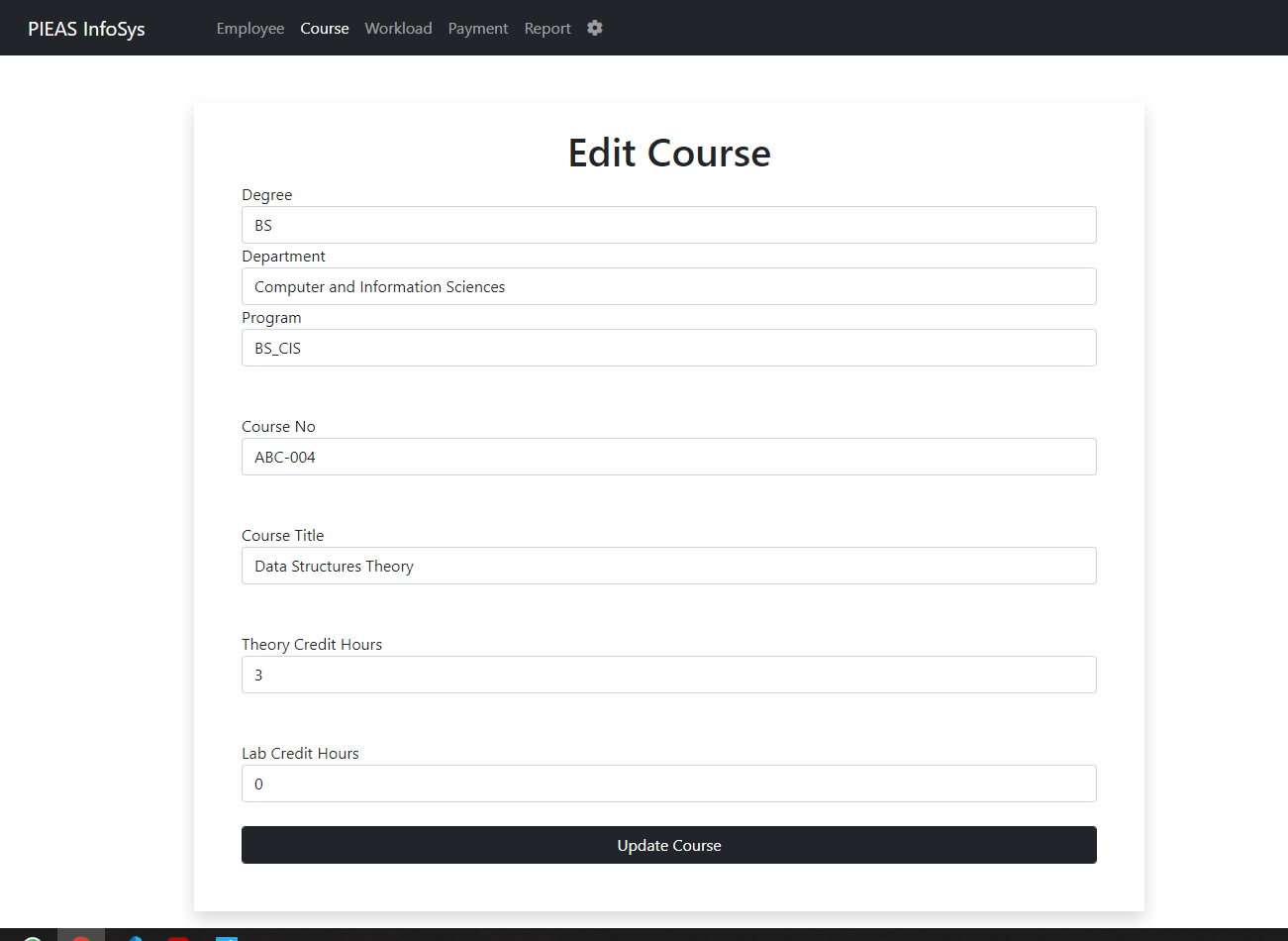
In this page user can add course along with attributes which all are necessary in this information system for example the number entered in course credit hours play an important part in calculating workload and total number of classes etc. The user interface of this page is shown in figure 4.12.



*Figure 4.12. Add Course Page*

### Edit Course Page

In this page, user can alter any information regarding course that is stored previously by the user. The user interface of this page is shown in figure 4.13.



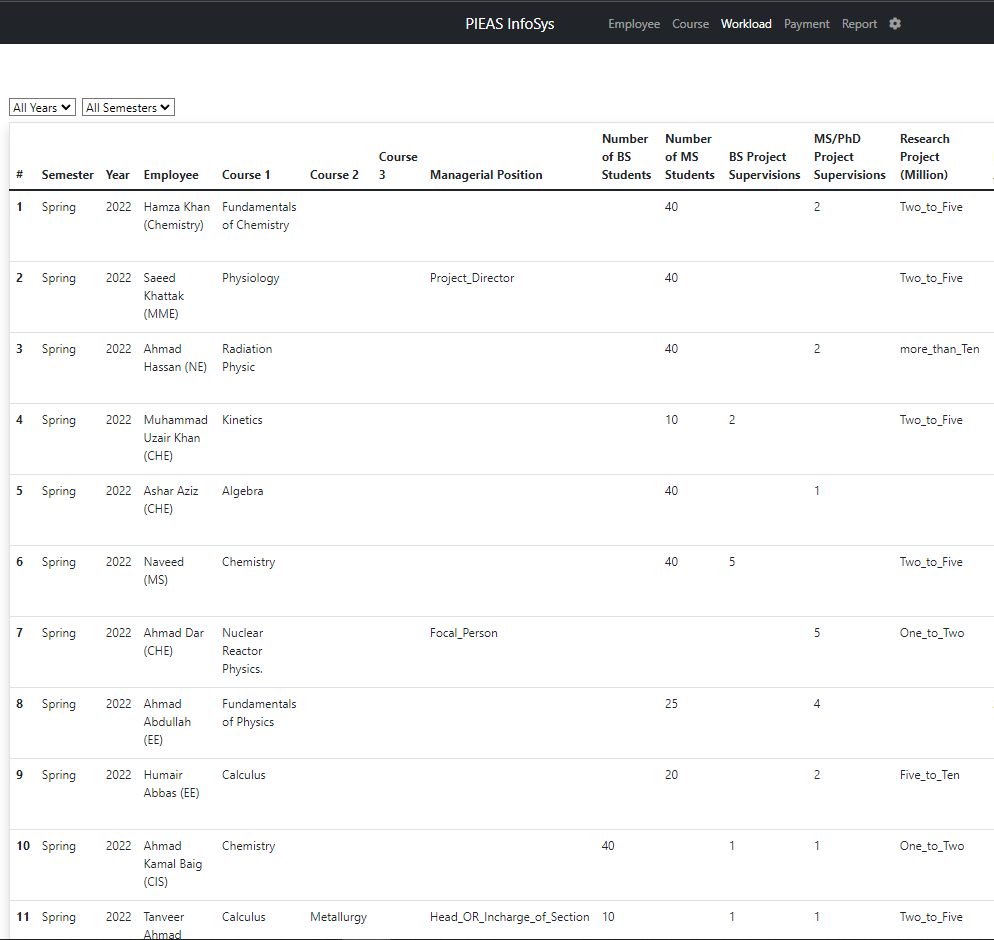
### Workload Page

*Figure 4.13. Edit Course Page*

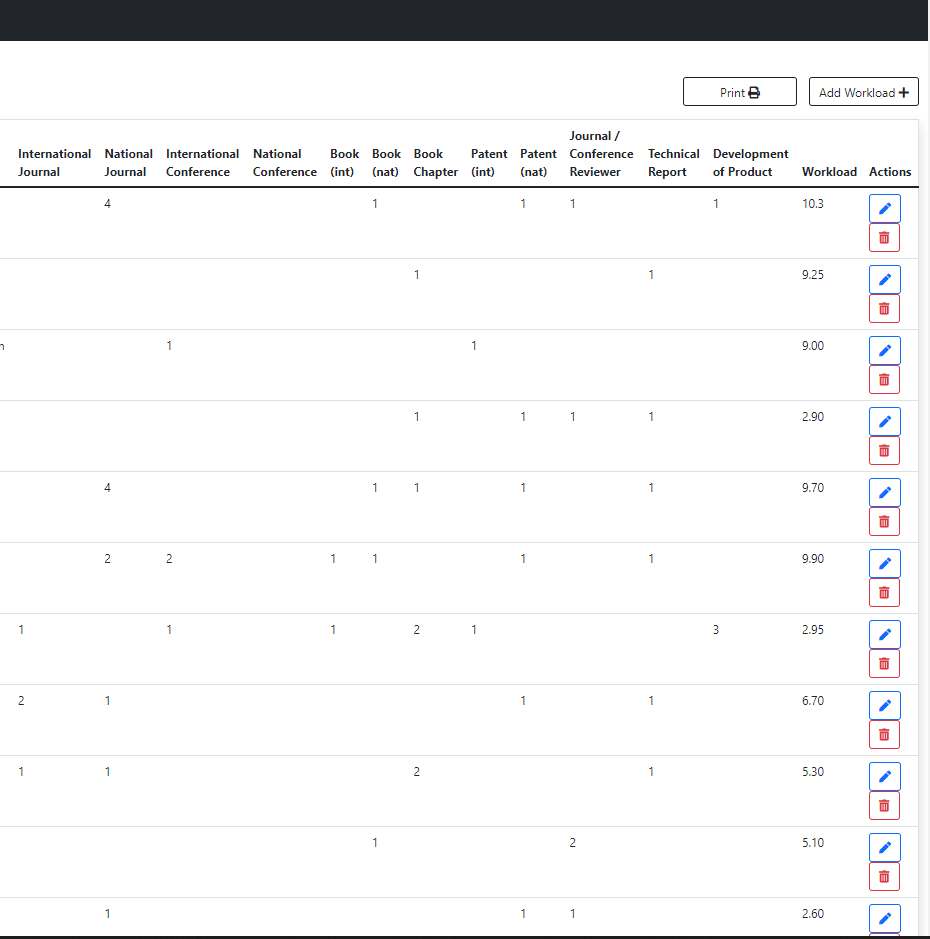
This module calculates the workload of every employee according to the credit hours acquired from the course management module. Other crucial data is also entered, like managerial position of a member, number of students a faculty member is delivering his/her lectures to and various other research and development factors; at the workload module.

This page also shows all the data regarding workloads of the employees. This workload will be calculated just after a form is filled in the “Add Workload” page. Workload is calculated using following attributes

* Semester
* Year
* Employee Name
* Course 1
* Course 2
* Course 3
* Managerial Position
* Number of BS Students
* Number of MS/PhD Students
* BS Project Supervisions
* MS/PhD Project Supervisions
* Research Project
* Publications
* Journal/Conf Reviewer
* Technical Report
* Book
* Patent
* Development of Product

The user interface of this page is shown in figure 4.14 (a) and (b).

*Figure 4.14(a) Workload Page*



*Figure 4.14(b) Workload Page*

### Add Workload Page

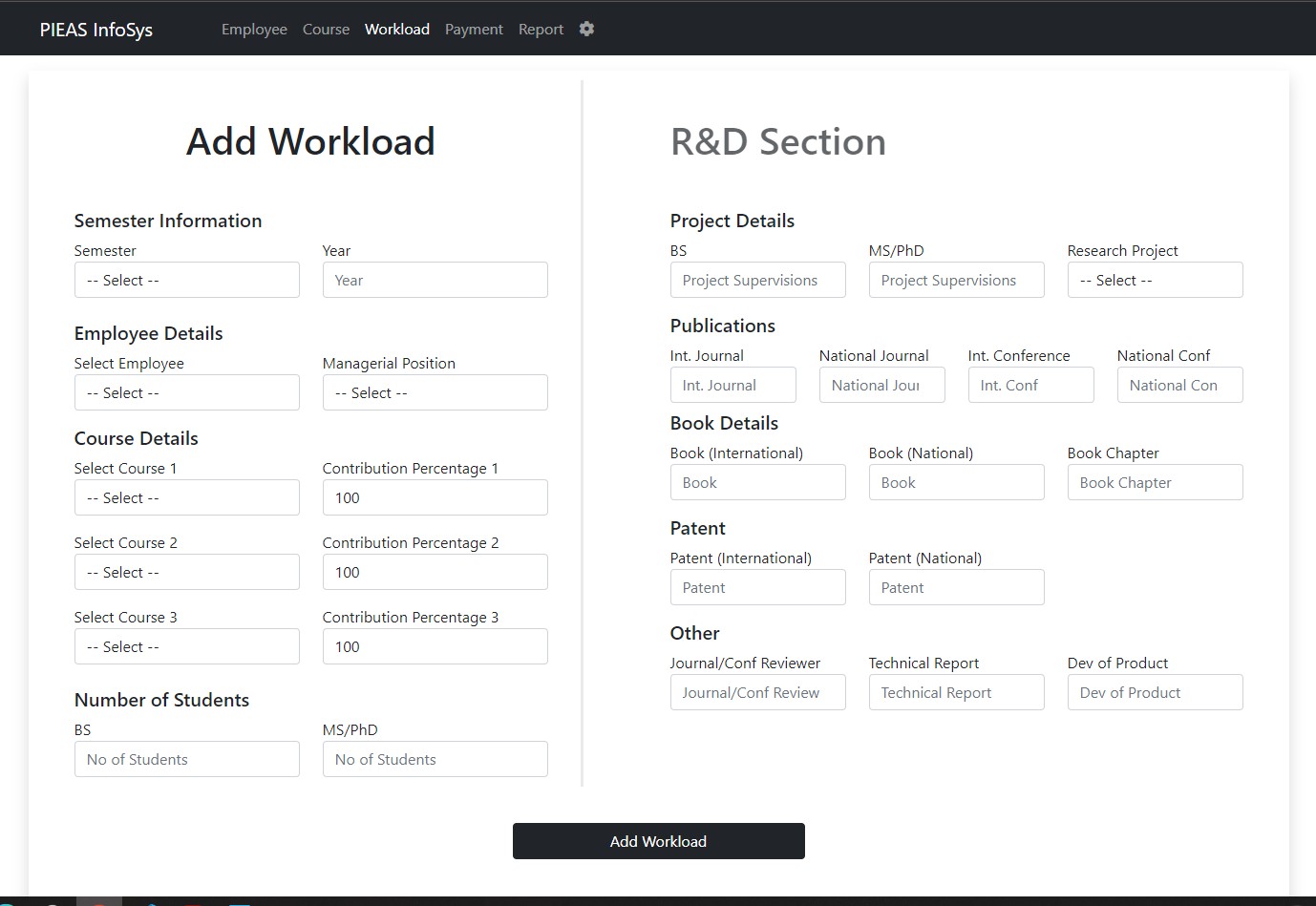
This page contains the form which is responsible for calculating workload score.

**Exact Formula of Workload Final Score**

Final Score =

[Theory CrHrs \* wDegree \* wTheory] + [Lab CrHrs \* wDegree \* wLab] + [wManagarial Position] + [No of BS Students \* wNoOfBsStudents] + [No of MS Students \* wNoOfMsStudents] + [BS Project Supervisions \* wBsProjectSupervisions] + [MS Project Supervisions \* wMsProjectSupervisions] + [wResearchProject] + [Int Journal \* wIJ] + [Nat Journal \* wNJ] + [Int Conf \* wIC] + [Nat Conf \* wNC] + [Journal/Conf Reviewer \* wJCR] + [ National Book \* wNBook] + [ International Book \* wIBook] + [Technical Report \* wTechnicalReport] + [Chapter \* wChapter] + [Dev of Product \* wDevOfProd] + [International Patent \* wIPatent] + [National Patent \* wNPatent]

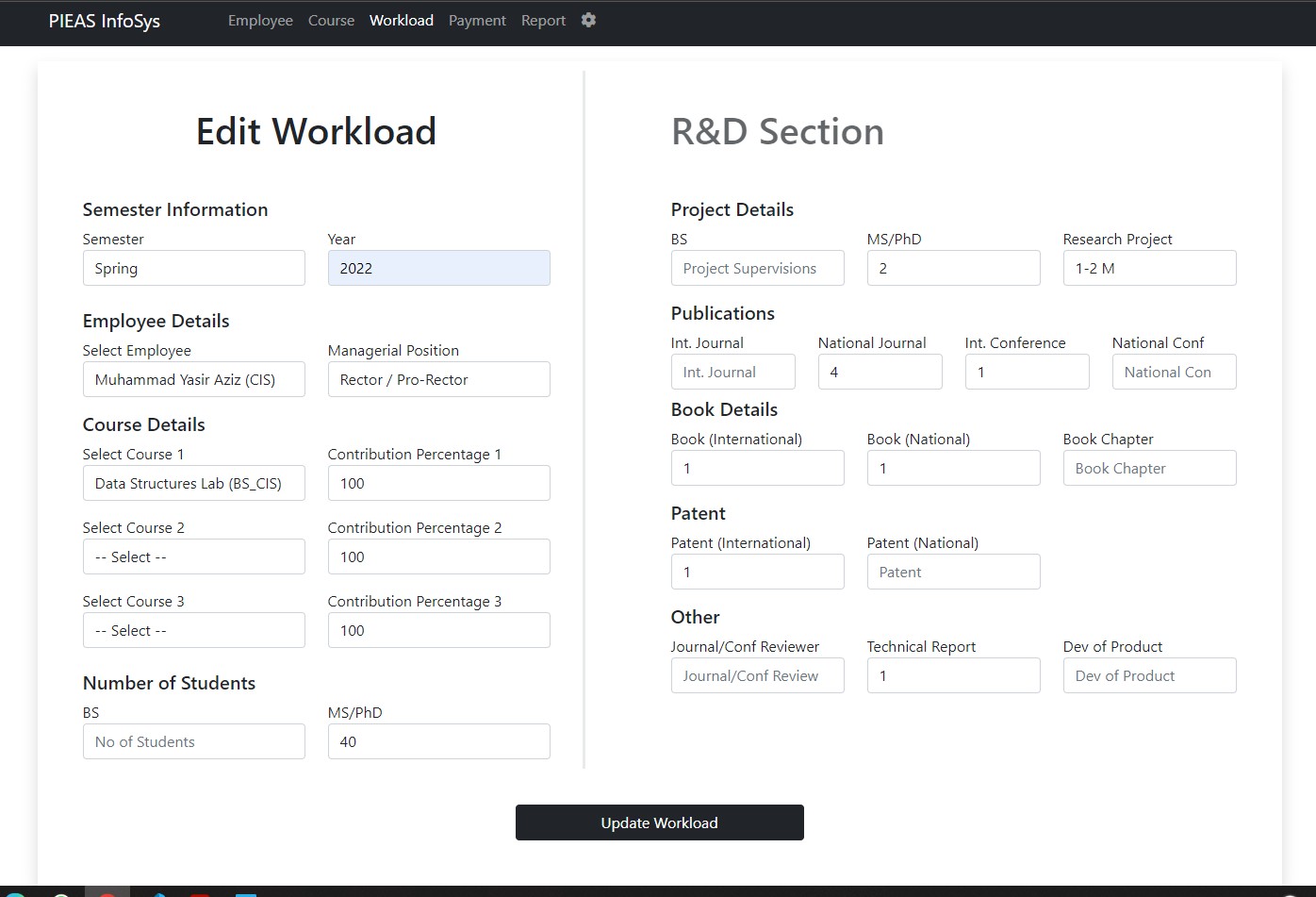
The user interface of this page is shown in figure 4.15.



*Figure 4.15. Add Workload Page*

### Edit Workload Page

This is the edit page to edit workload information. The user interface of this page is shown in figure 4.16.

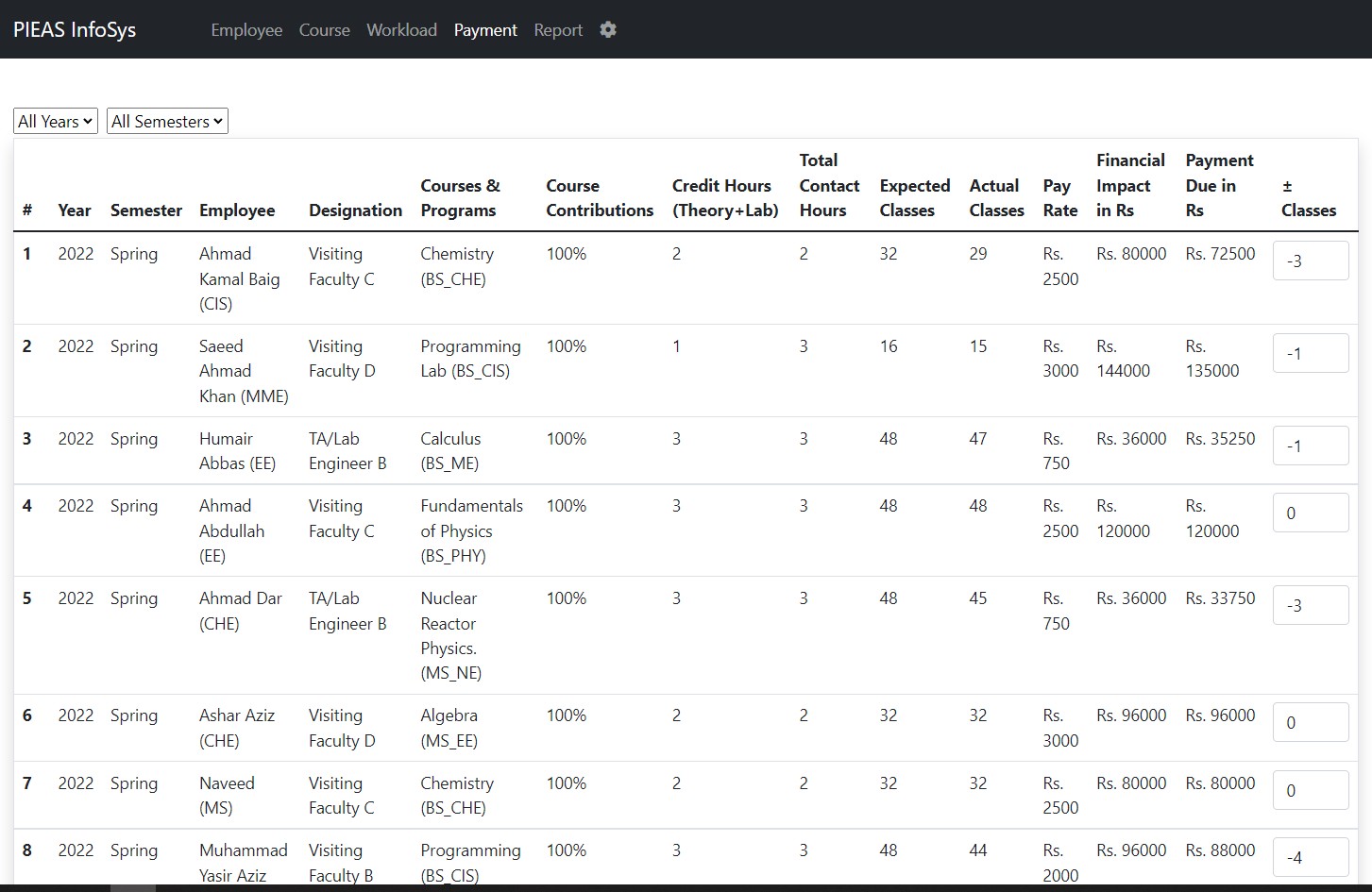


### Payment Page

*Figure 4.16. Edit Workload Page*

This page contains all the data of employees regarding payment. This page has functionality to calculate:

* + - 1. Financial Impact [Expected Classes \* Pay Rate \* Total Contact Hours]
      2. Payment Due [Actual Classes \* Pay Rate \* Total Contact Hours] The user interface of this page is shown in figure 4.17.



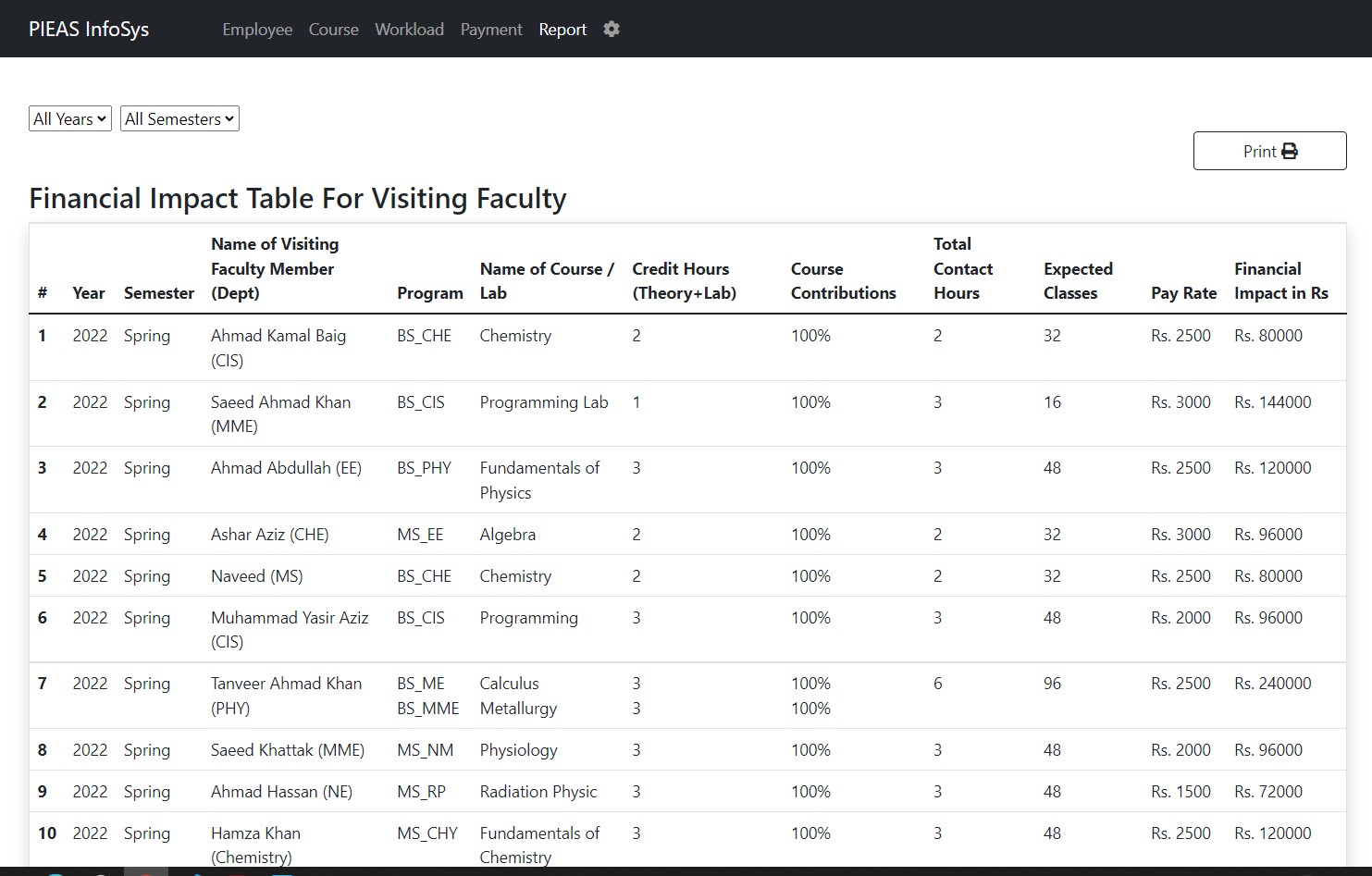
### Report Page

*Figure 4.17. Payment Page*

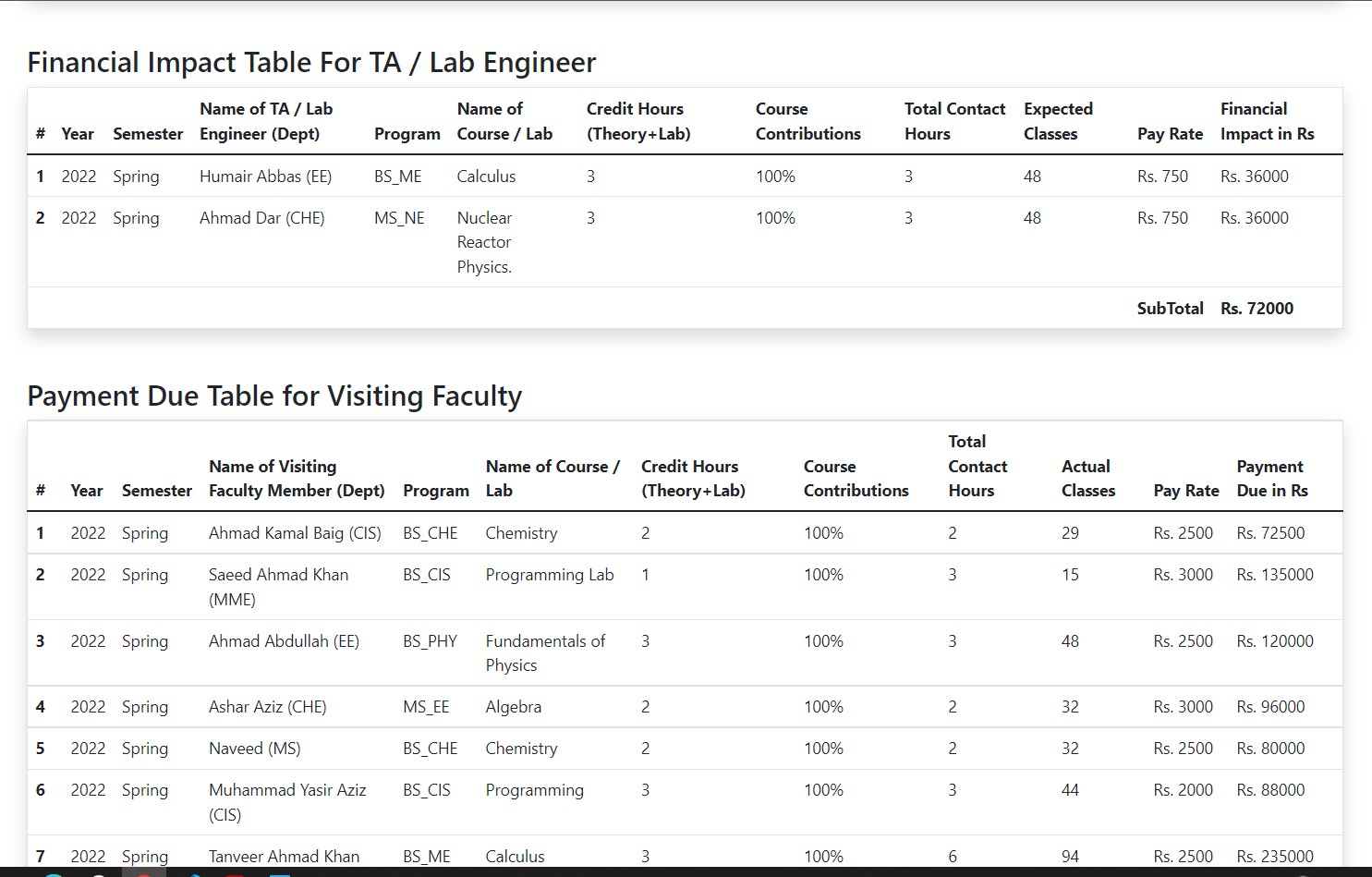
In this page a final report is generated which contains following tables:

* + - 1. Financial Impact Table for Visiting Faculty
      2. Financial Impact Table for TA/Lab engineer
      3. Payment Due Table for Visiting Faculty
      4. Payment Due Table for TA/Lab Engineer
      5. Department Wise Financial Impact Summary
      6. Department Wise Payment Due Summary

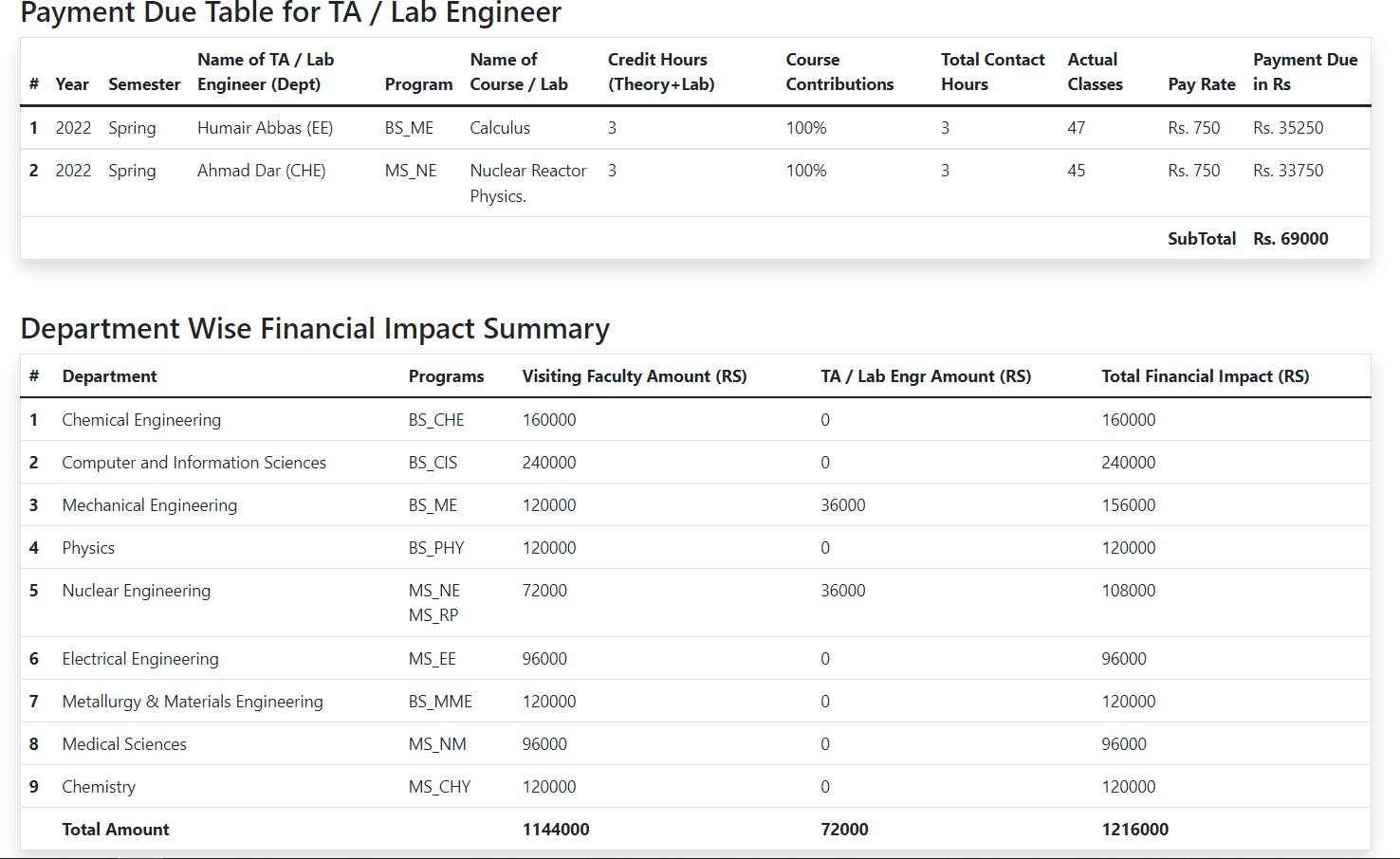
The user interface of this page is shown in figure 4.18 (a), (b), (c) and (d).



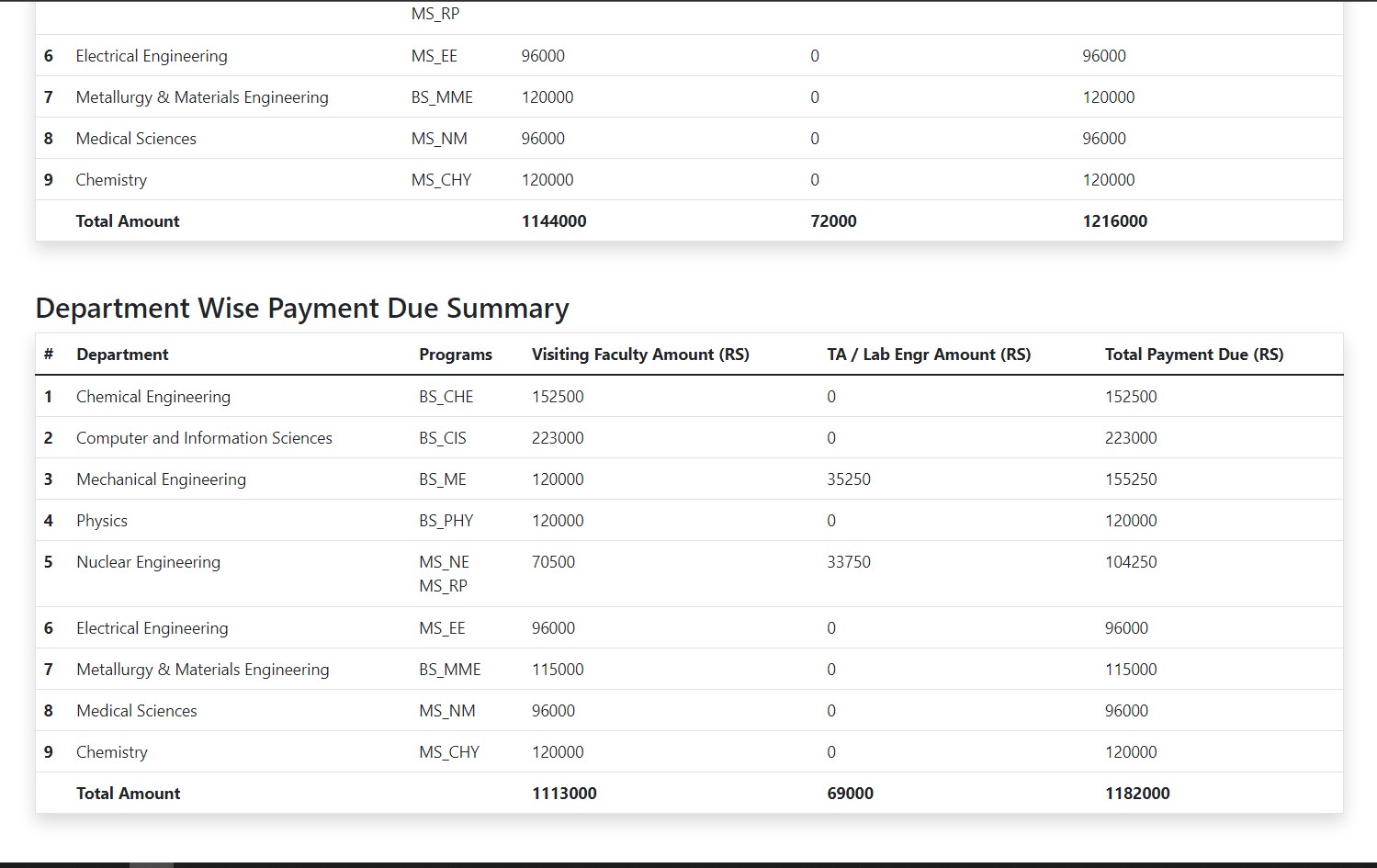
*Figure 4.18(a) Report Page*



*Figure 4.18(b) Report Page*



*Figure 4.158(c) Report Page*



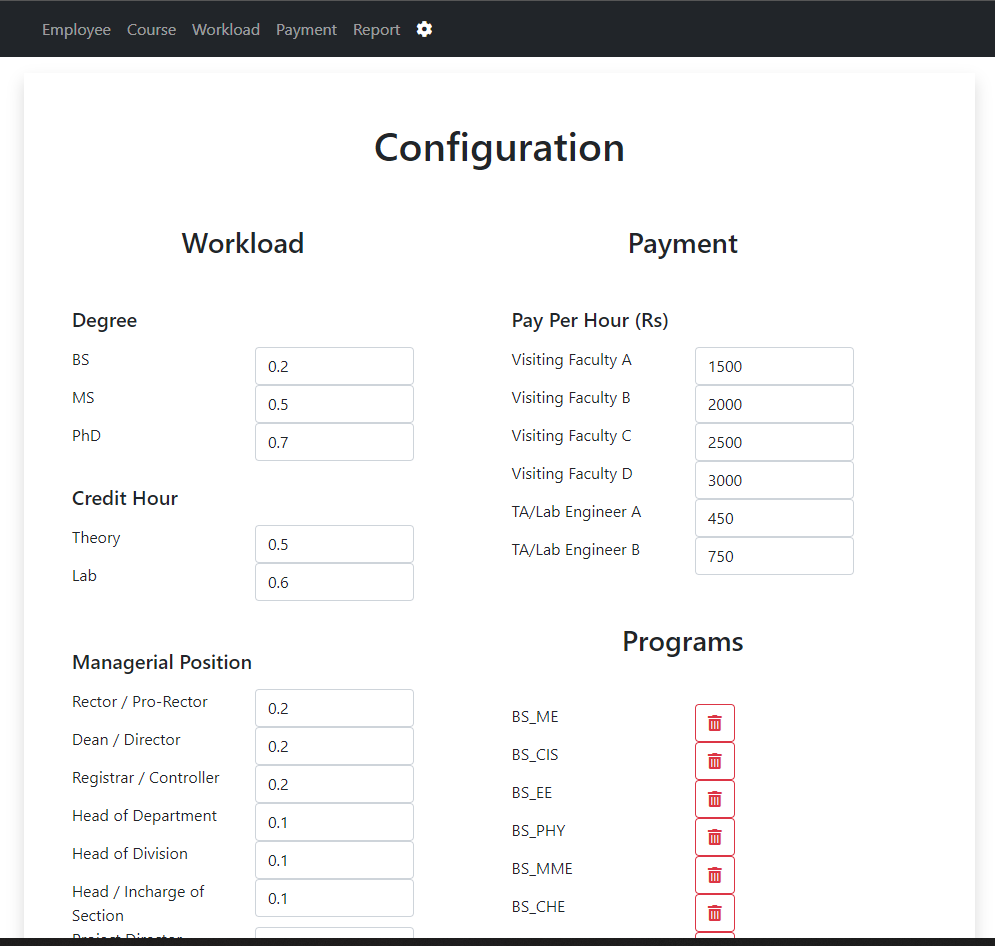
*Figure 4.18 (d) Report Page*

### Configuration Page

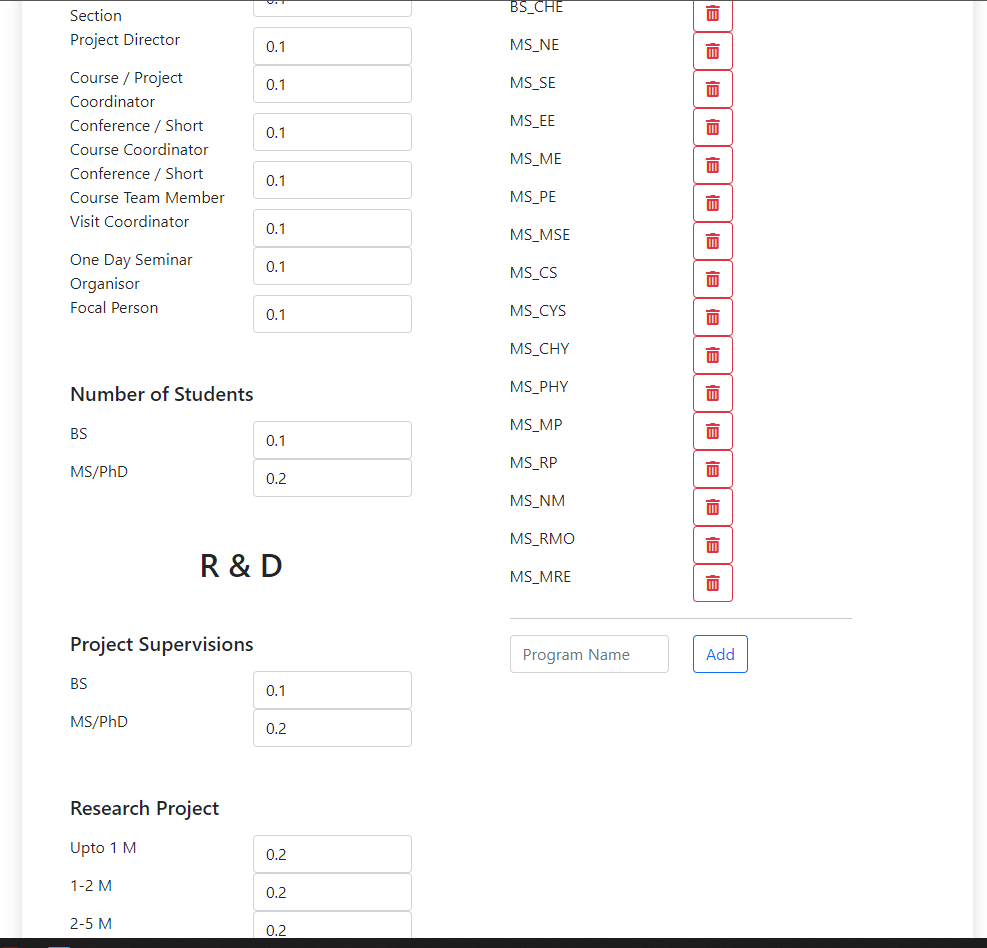
This is used for:

* + - 1. Setting weights of the factors that are involved in calculating final score of workloads.
      2. Changing pay per hour that is corresponding to the designation of an employee.
      3. Add/Delete programs in the information system.

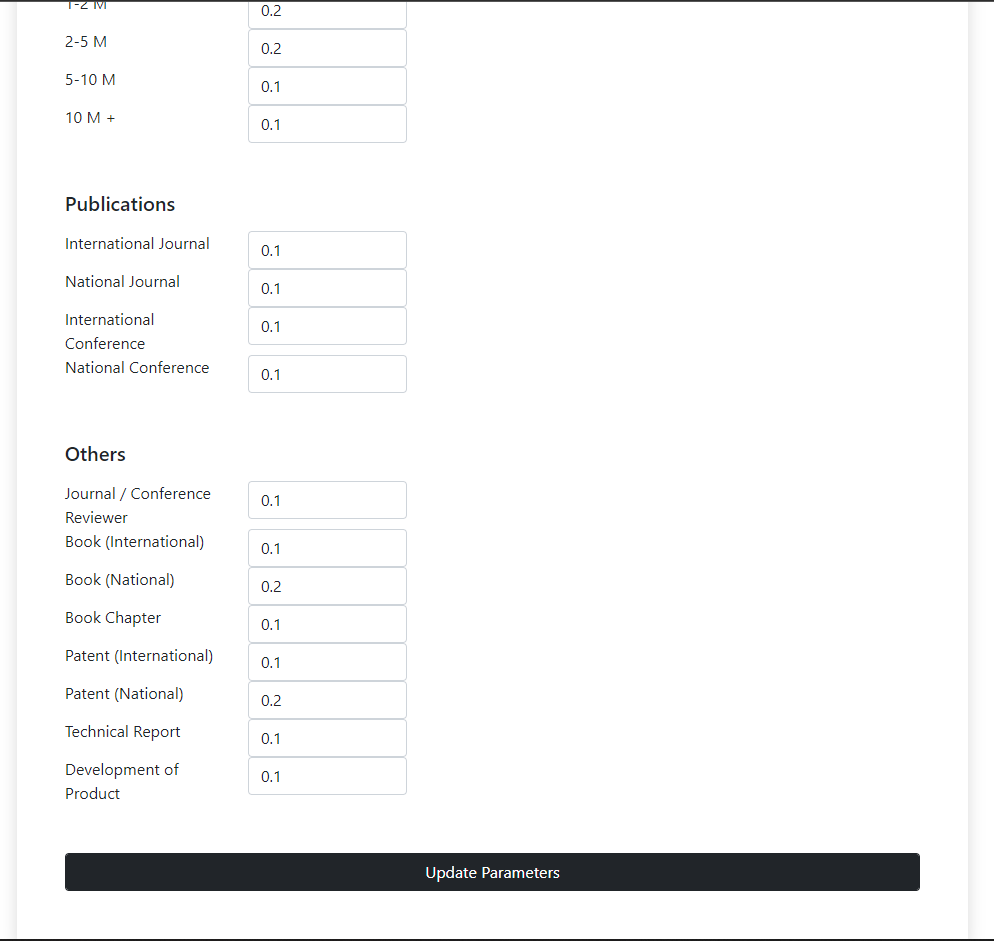
The user interface of this page is shown in figure 4.19 (a), (b) and (c).



*Figure 4.19(a) Configuration Page*



*Figure 4.19(b) Configuration Page*



*Figure 4.19(c) Configuration Page*

**Chapter 5: Conclusion and Future Recommendations**

* 1. **Conclusion**

A good page design is essential for modern Web development. A good page layout should generally satisfy the fundamental components of a good page design. This comprises the use of consistent visuals, color contrast, text arrangement, font choice, page layout, and size. In order to design a page well for a certain audience, the demographics and history of the users must be organized. Although, it is be hard to come up with such a design that is well suited to all of the users of the system.

Beginning with the very fundamentals of HTML, CSS, and the JavaScript. I have progressed through several areas to demonstrate how to use these languages for content structuring and presentation. The open nature of the Internet and the fact that all content is supplied as documents to your web browser, enabling you to read other people's source code, substantially helps in building our own project.

The “Information System for Managing Visiting Faculty and TA/Lab Engineers” design and development was challenging, and the project's documentation required a lot of effort and time. However, with perseverance and hard work, nearly all of the challenges were solved, and a usable software product was created. After the creation of this information system, development of any type of information system is now quite simple.

* 1. **Future Recommendations**

The system “Information System” has almost all the basic functionalities which are required to help users, but there is still a need for new functionality that is “User Management”. Another functionality that should also be introduced in the proposed system is that the system can also calculate net payment of employees and generate a report if one employee is engaged in more than one courses. After adding this functionality, the system should go through testing phase and then finally, it should head towards deployment.

A multi-purpose web application should use a graphics navigation bar with rollover effects, in my opinion, after carefully examining all the information that has been researched and user input. Each page should feature a text-only navigation bar for ease of use in addition to the graphic navigation bar with rollover effects. To provide a uniform appearance and feel across the site, the developer should also establish a Cascading Style Sheet to regulate the typeface and font size.

Some SOPs that must be considered before using the system:

* + 1. There must be one admin account pre-created in the database to use the system.
    2. Only admin or course coordinator have access to add, update or delete any entry.
    3. No one will be able to delete the above-mentioned admin account.
    4. The current logged in user cannot delete his account.
    5. There can be any number of users of the system.
    6. These users will have access to the system according to their role.

# References

1. “HTML | PyCharm,” *PyCharm Help*. [https://www.jetbrains.com/help/pycharm/editing-](https://www.jetbrains.com/help/pycharm/editing-%20%20%20%20%20html-files.html) [html-files.html](https://www.jetbrains.com/help/pycharm/editing-%20%20%20%20%20html-files.html) (accessed Aug. 15, 2022).
2. “Returning To Plain HTML + CSS. Over the past week, I tried my hand at… | by Danny Krug | Medium.” [https://medium.com/@dannykrug/returning-to-plain-html-css-](https://medium.com/%40dannykrug/returning-to-plain-html-css-4386b61f3da) [4386b61f3da](https://medium.com/%40dannykrug/returning-to-plain-html-css-4386b61f3da) (accessed Aug. 18, 2022).
3. “React – A JavaScript library for building user interfaces.” <https://reactjs.org/> (accessed Aug. 28, 2022).

[4].“Databases and Collections — MongoDB Manual.” <https://www.mongodb.com/docs/manual/core/databases-and-collections/> (accessed Aug. 29, 2022).

1. “The Complete 2022 Web Development Bootcamp,” *Udemy*[.](https://www.udemy.com/course/the-complete-web-development-bootcamp/) <https://www.udemy.com/course/the-complete-web-development-bootcamp/> (accessed Aug. 16, 2022).
2. M. O. contributors Jacob Thornton, and Bootstrap, “Bootstrap.” <https://getbootstrap.com/>(accessed Aug. 24, 2022).
3. “MongoDB Atlas: Cloud Document Database | MongoDB.” [https://www.mongodb.com/cloud/atlas/lp/try2?utm\_content=controlhterms&utm\_source](https://www.mongodb.com/cloud/atlas/lp/try2?utm_content=controlhterms&utm_source=google&utm_campaign=gs_emea_pakistan_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0dXbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw_wcB)

[=google&utm\_campaign=gs\_emea\_pakistan\_search\_core\_brand\_atlas\_desktop&utm\_ter](https://www.mongodb.com/cloud/atlas/lp/try2?utm_content=controlhterms&utm_source=google&utm_campaign=gs_emea_pakistan_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0dXbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw_wcB) [m=mongodb&utm\_medium=cpc\_paid\_search&utm\_ad=e&utm\_ad\_campaign\_id=12212](https://www.mongodb.com/cloud/atlas/lp/try2?utm_content=controlhterms&utm_source=google&utm_campaign=gs_emea_pakistan_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0dXbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw_wcB) [624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0d](https://www.mongodb.com/cloud/atlas/lp/try2?utm_content=controlhterms&utm_source=google&utm_campaign=gs_emea_pakistan_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0dXbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw_wcB) [XbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw\_wcB](https://www.mongodb.com/cloud/atlas/lp/try2?utm_content=controlhterms&utm_source=google&utm_campaign=gs_emea_pakistan_search_core_brand_atlas_desktop&utm_term=mongodb&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624545&adgroup=115749718983&gclid=Cj0KCQjwr4eYBhDrARIsANPywChc7Rl0dXbU5oNRLKUhiUPNuxtg1suhu9H0H8seDK0506n9W24dCKAaAp4CEALw_wcB) (accessed Aug. 24, 2022).

1. “W3Schools Free Online Web Tutorials.” <https://www.w3schools.com/>(accessed Aug. 25, 2022).
2. “MDN Web Docs.” [https://developer.mozilla.org](https://developer.mozilla.org/) (accessed Aug. 25, 2022).
3. “Database Management System”, published by IT Series Publications and the authors fo the book are Imran Saeed, Tasleem Mustafa, Tariq Mahmood and Ahsan Raza Sattar. [https://freebooks.pk/books/database-management-system](https://freebooks.pk/books/database-management-system/)/ (accessed Aug. 26, 2022).
4. “Entity Diagram - diagrams.net.” <https://app.diagrams.net/> (accessed Aug. 29, 2022).

**Appendix A – Instructions to Use Information System**

Instructions to use Information System:

* To use the system, a user must have an account and must be logged in the system.
* All the configuration related to programs, weights and pay rates will be on the configuration page.
* All the employees and courses should be added and checked before filling the workload page.
* In the course section, while adding courses, if the course has theory credit, lab credit must be filled with zero and vice versa.

# Appendix B – React JS Installation and Setup

1. **Install Nodejs**

Packages for React are managed and shared using NPM, the Node package manager. Along with Nodejs, NPM will also be installed. It is possible to download and set up Node.js from the official Node Js website.

https://nodejs.org

Once the Installation of Node is complete. Open Node Js Command Prompt and we can check the version as well.

1. **Install Create-React-App Tool**

The following action is to use NPM to install a project called create-react-app. Utilizing our technology, this tool makes it simple to construct React applications. This can be set up either permanently at the system level or momentarily at a folder level.

npm install -g create-react-app

1. **Creating a New React Project**

After installing create-react-app, we can start building our first react application. Following is command used to create a new project

create-react-app name-of-project

Don't use an uppercase character while creating the project.

1. **Running the React Application**

Now lets run it locally on our system using npm start command. Launch the browser and visit http://localhost:3000. We can see our first React Application in the browser.

cd name-of-project npm start

Now, we have created a new project using react and executed the project.

1. **Install Visual Studio Code**

Download and install Visual Studio Code from the following URL

<https://code.visualstudio.com/download>

After the VS Code installation, open the project that has been created earlier using the VS Code. The Project has the following 3 folders

* 1. Node\_modules
  2. Public
  3. src

The output we have seen when the project is executed comes from a file called Index.html which resides inside the public folder.

# Appendix C – Libraries Used in Project

Name and purpose of libraries used in this project:

### Bootstrap

This is a CSS framework that is used to create responsive websites.

* **React**

This is a front-end java script library used to build user interfaces.

* **Axios**

This library is used to make http requests from frontend.

* **React-Router-Dom**

This library is used to make routes in frontend.