FACULTY PROFILE MAINTENANCE

A PROJECT REPORT

Submitted by

ARJUN P (950018104007)

KANNAN B (950018104017)

MAHESH VARUN R (950018104025)

VIGNESH S (950018104054)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS-TIRUNELVELI

ANNA UNIVERSITY: CHENNAI 600025

JUNE 2022

FACULTY PROFILE MAINTENANCE

A PROJECT REPORT

Submitted by

ARJUN P (950018104007)

KANNAN B (950018104017)

MAHESH VARUN R (950018104025)

VIGNESH S (950018104054)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS-TIRUNELVELI

ANNA UNIVERSITY: CHENNAI 600025

JUNE 2022

ANNA UNIVERSITY: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project **"FACULTY PROFILE MAINTENANCE"** is a bonafide work of **"ARJUN P (950018104007), KANNAN B (950018104017), MAHESH VARUN R (950018104025), VIGNESH S (950018104054)"** who carried out the project under my supervision.

Signature of the HOD

Dr. K. Saravanan,

Assistant Professor,

Department of Computer Science

and Engineering,

Anna University Regional Campus

-Tirunelveli,

Tirunelveli – 627 007

Signature of the Supervisor

Dr. E. Golden Julie,

Assistant Professor,

Department of Computer Science

and Engineering,

Anna University Regional Campus

-Tirunelveli,

Tirunelveli – 627 007

Submitted to CS-8811 – Project work – Viva Voce – held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, we thank Almighty for enabling us to complete our project work successfully.

We wish to express our deep sense of gratitude to our Dean **Dr. N. SHENBAGA VINAYAGA MOORTHI,** Dean, Anna University Regional Campus – Tirunelveli, for his support and encouragement throughout this project work.

We express our sincere thanks to our Head of the Department **Dr. K. SARAVANAN,** Assistant Professor, Department of Computer Science and Engineering, for his motivation, inspiration and encouragement to undertake this work.

We express our sincere thanks to our project coordinators **Mrs. T. KAVITHA**, Assistant Professor, Department of Computer Science and Engineering, for his giving such attention and time in steering the project.

We owe our sincere thanks and gratitude to our project guide **Dr. E. GOLDEN JULIE,** Assistant Professor, Department of Computer Science and Engineering for her suggestions and support during the course of our project work.

We are also indebted to all the teaching and non-teaching staff members of our college for helping us either directly or indirectly by all means throughout the course of our study and project work. We wish to thank our family and friends for being there always in our course of life

ARJUN P

KANNAN B

MAHESH VARUN R

VIGNESH S

ABSTRACT

This project is named “**FACULTY PROFILE MAINTENANCE”.** The main purpose of our project is to view faculty details and information in a form of pdf. The faculty information pdf can be viewed by anyone who may be a faculty or a student in the institution. This project has been done for our institution Anna University Regional Campus – Tirunelveli. Using this college website people from anywhere can view the faculty members information who works in the institution. Not only faculty member as teaching staff but also can view non-teaching staff members who work in the institution. New user can register on the register page and after they registered a mail id had been sent to the mail account in which they have to verify the account using the mail account. After they registered and verified, they can go login to the login page and login to their account and can enter the information about them in the edit page. If they have any changes in their information and details the can choose the edit button and can change their information and if they finished, they can click the next button and add their information. Last, they can submit the form and their information will store in the database and their name will be displayed in the home page. If they click view, they can view their details in the form of pdf and can download the pdf. This project will be used by our faculty members who works in the institution. The admin who logged in through the admin log in page can generate public and private data as report.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

HTML	Hyper Text Mark-up Language
JS	Java Script
CSS	Cascading Style Sheet
PHP	Hypertext Pre-processor
RAM	Random Access Memory
IDE	Integrated Development Environment
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
DTD	Data Flow Diagram
ER	Entity Relationship

CHAPTER – 1

INTRODUCTION

The project title is **“Faculty Profile Maintenance”**. The purpose of the project is to view the details and the profile of our faculty members of our organizations teaching and non-teaching staff who works in the institution. It also provides the functionality of downloading information’s or forms in a form of pdf for printing purpose. The Faculty Profile Maintenance also contains a report session that can only be accessed by the admin. Also, the admin has a separate login page. The report contains all the private and public data and the report can be filtered for a particular period of time. The project is a web-based system available for any workstation through the browser. The website is easy to use and has a smooth and better flow on the web page. Multiple authentication (read-only, faculty, admin) are there in our project. The web portal “Faculty Profile Maintenance” is made by using php and xampp server. Our website is useful for all the faculty members to view and update their profile. The main benefit of our project is that anybody from anywhere can view the profile of the faculty who works in the institution. Our project is also responsible for any mobile devices

1.1 OBJECTIVE:

The purpose of the project is to build an application to reduce the manual work for managing faculty and employee information. Easy to filter the faculty details. It tracks all the details about the faculty who works in the institution.

1.2 PROBLEM DEFINITION

A Faculty Information Systems is a software designed webpage to simplify the Collection, management, analysis and use of information about the faculty members in an institution

- Transparent tracking of faculty data
- Improved fairness across the institution
- Better experience for the user
- Easier to find relevant information

1.3 LITERATURE REVIEW

Anna University faculty Profile (AUKDC Faculty Profile) :

Link : https://www.aukdc.edu.in/fis/facprofile_campus

- This webpage came live in the mid of 2019
- They used this webpage to show their profile of the faculty who works in the institution
- The faculty members can edit their information and download their information as a form of pdf

CHAPTER – 2

SYSTEM DESCRIPTION

2.1 SYSTEM REQUIREMENTS

2.1.1 SOFTWARE REQUIREMENTS

- Front end: HTML, CSS, Bootstrap, JavaScript.
- Back-end: PHP, MySQL.
- Operating System: Windows, Linux, Mac
- Code Editor: VS Code.
- Web Browser: Chrome, Firefox, etc.
- Web Server: XAMPP

2.1.2 HARDWARE REQUIREMENTS

- Processor: Intel core i3 or higher
- Processor Speed: 1.8 GHz
- RAM: 2GB
- Storage: 1GB of free hard drive space.
- System Type: 32-bit Operating System.

2.1.3 FEATURES OF THE SOFTWARE

2.1.3.1 PHP

PHP (Hypertext Pre-processor) is a widely-used opensource general-purpose scripting language that is especially suited for web development and can be embedded into HTML

PHP is a general-purpose scripting language geared toward web development. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Pre-processor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

2.1.2.2 HTML

The **Hypertext Markup Language** or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as **Cascading Style Sheets (CSS)** and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as **<img />** and **<input />** directly introduce content into the page. Other tags such as **<p>** surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. A form of HTML, known as HTML5, is used to display video and audio, primarily using the <canvas> element, in collaboration with JavaScript.

2.1.3.3 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

2.1.3.4 MYSQL

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, php, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, scalable, and easy to use
- MySQL is compliant with the ANSI SQL standard
- MySQL is named after co-founder Monty Widenius's daughter: My

MySQL supports large database, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase it to a theoretical limit of 8 million Terabytes (TB)

2.1.3.5 JAVASCRIPT

JavaScript is the world's most popular programming language. JavaScript is the programming language of the Web. JavaScript Can Change HTML Content. One of many JavaScript HTML methods is getElementById().Over 97% of websites use JavaScript on the client side for web page behaviour, often incorporating third-party libraries.

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs that are specific to Web pages, please see Web APIs and DOM.

2.1.3.6 BOOTSTRAP

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.It is a front-end framework used for easier and faster web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Initially, Bootstrap was called Twitter Blueprint and was developed by a team working at Twitter. It supports responsive design and features predefined design templates that you can use out of the box, or customize for your needs with your code. You don't need to worry about compatibility with other browsers either, as Bootstrap is compatible with all modern browsers and newer versions of Internet Explorer.

2.1.3.7 Visual Studio Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python, C++ and Fortran. It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Out of the box, Visual Studio Code includes basic support for most common programming languages. This basic support includes syntax highlighting, bracket matching, code folding, and configurable snippets. Visual Studio Code also ships with IntelliSense for JavaScript, TypeScript, JSON, CSS, and HTML, as well as debugging support for Node.js. Support for additional languages can be provided by freely available extensions on the VS Code Marketplace.

2.1.3.8 XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP

Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as WordPress and Joomla! can also be installed with similar ease using Bitnami.

2.2 System Analysis

2.2.1 Existing System:

Initially colleges maintain the profiles of the faculty members to view the details and information about the faculty. Anyone can view the faculty profile

Drawbacks in Existing system:

i. **Difficulty in modifying the changes:**
   - In the existing system the faculty can only upload the completed details in a form of pdf. The faculty cannot change any details or information in the uploaded pdf regarding the changes.

ii. **No admin Login:**
   - There is no admin login in the existing system

iii. **Generate Report:**
   - The admin in the existing system or website cannot generate the report, where the report contains the information about the private and public data of the faculty who is in the institution and the report can be filtered for a particular period of time

iv. **No private data:**

- There is no private data in the existing system.
- In the existing system the private data cannot be viewed by neither the faculty nor the admin.
- The private data contains confidential information about the work of the faculty

2.2.2 Proposed System:

This is a web-based application that allows to view the profile of

the faculty members who work in the institution

2.2.2.1 Advantages

i. **Modify the data:**

- In our new system after the faculty logged in, they can add their details and information. If there is any information regarding the changes in the information, they given they can modify or edit the information. If they want to remove the information, they can remove the information by the delete option given in our system.

ii. **Generation of pdf:**

- The pdf will be generated automatically for the information they gave earlier there will be no need for uploading any pdf. The pdf will be generated automatically by clicking the option generate pdf.

iii. **Admin Login:**

- A new page of admin login is created in the new system, where the old system does not contain any login for admin.

iv. Report Generation:

- The report generation in the new system can only be accessed by the admin. The admin cam filter the details of the information about the faculty members data by the filter option given in the new system.

v. Separation of data

- There is a separation of private and public data in the new system. The private data can only be accessed by the admin and the particular faculty who gave the information or details about them. The private data cannot be accessed by other people whereas the public data can be accessed by all the people.

CHAPTER – 3

SYSTEM DESIGN

3.1 INTRODUCTION

System design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be build. Beginning, once system requirements has been specified an analysed, system design is the first three technical activities – design, code and test is required to build and verify software.

The importance can be stated with a single word “QUALITY”, design is the fostered in software development. Design provides us with representation of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished product or system. Software design servesas a foundation foe=r all software engineering steps that follow, without a strong design we risk n=building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of the data structure, program structure and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

3.2 DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.

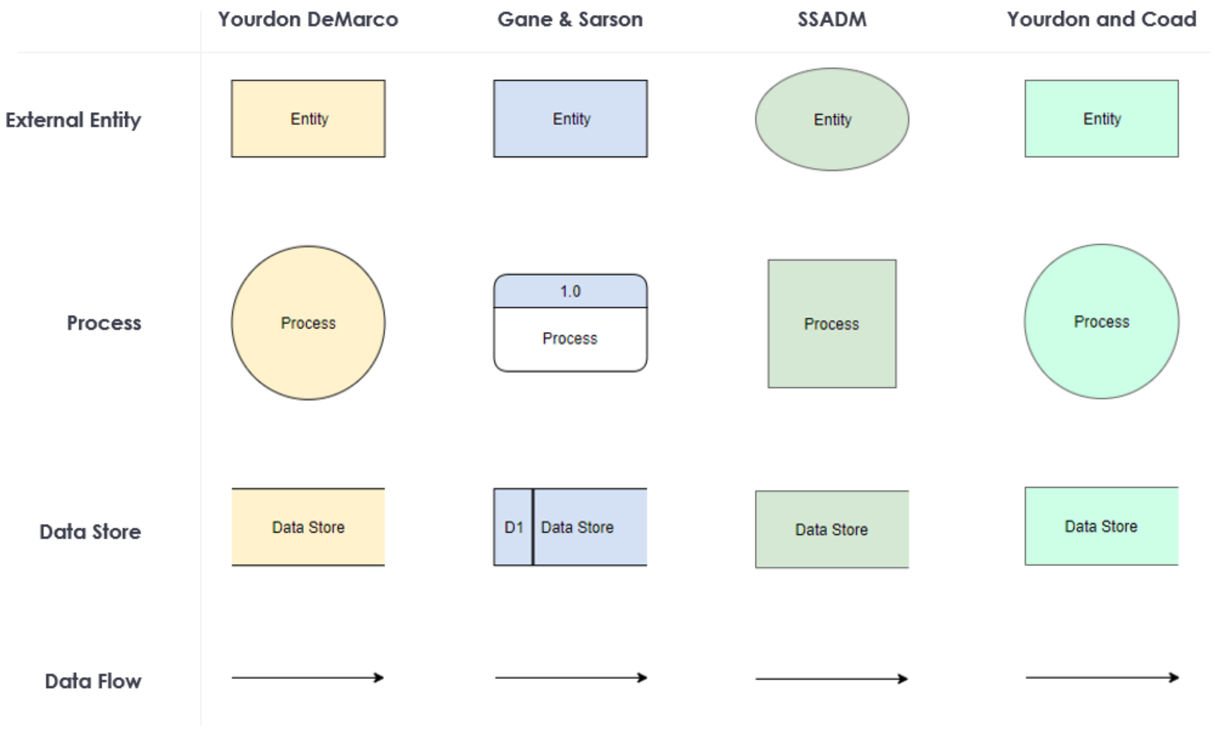
All data flow diagrams include four main elements: entity, process, data store and data flow.

External Entity – Also known as actors, sources or sinks, and terminators, external entities produce and consume data that flows between the entity and the system being diagrammed. These data flows are the inputs and outputs of the DFD. Since they are external to the system being analysed, these entities are typically placed at the boundaries of the diagram. They can represent another system or indicate a subsystem.

Process – An activity that changes or transforms data flows. Since they transform incoming data to outgoing data, all processes must have inputs and outputs on a DFD. Processes are typically oriented from top to bottom and left to right on a data flow diagram.

Data Store –. Data stores could consist of files held long term or a batch of documents stored briefly while they wait to be processed. Input flows to a data store include information or operations that change the stored data. Output flows would be data retrieved from the store.

Data Flow – Movement of data between external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow. This data could be electronic, written or verbal. Input and output data flows are labelled based on the type of data or its associated process or data store, and this name is written alongside the arrow.



DATA FLOW DIAGRAM SYMBOLS

3.2.1 DFD LEVEL 0

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

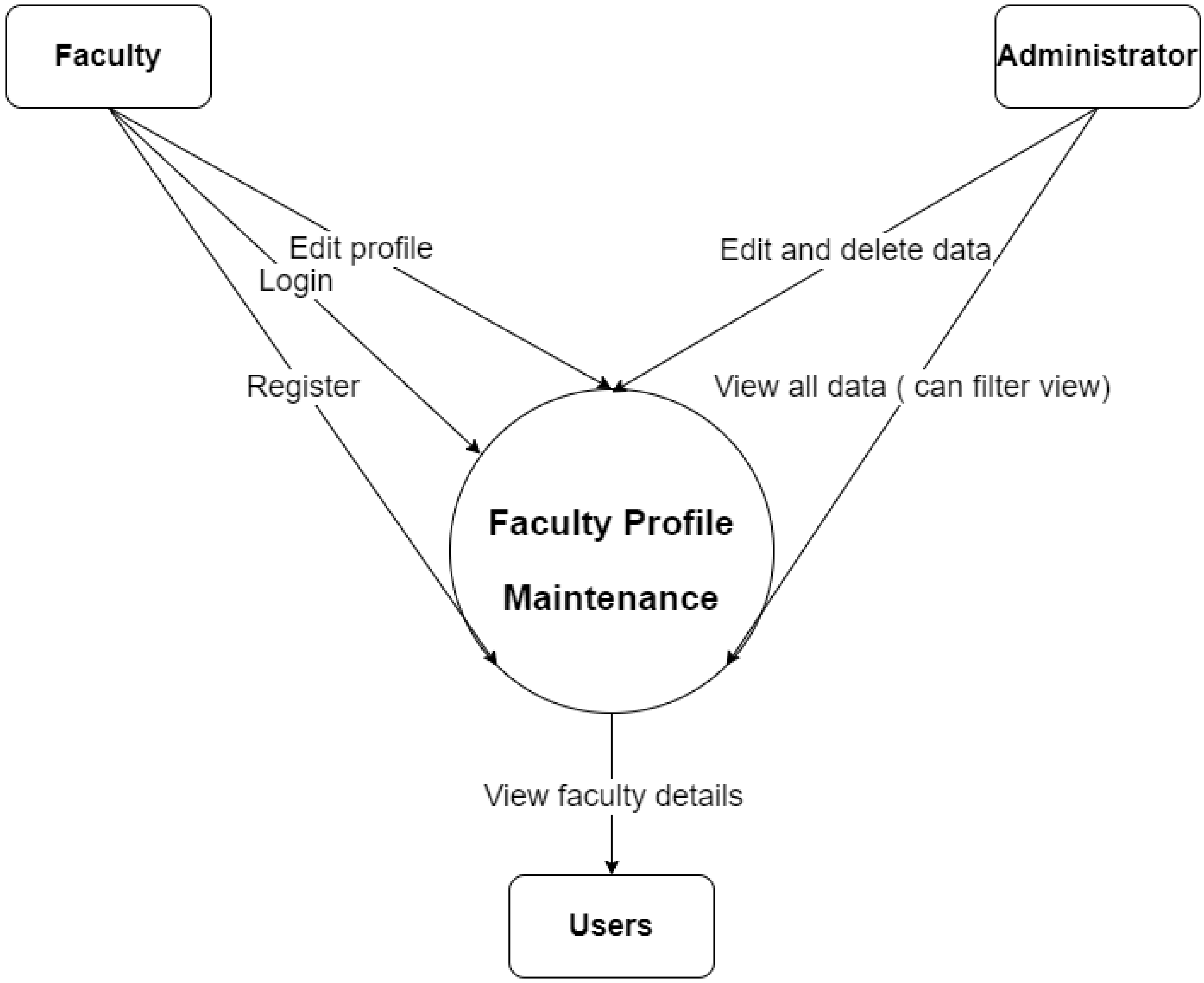


Fig 3.2.1 level 0 DFD for “faculty profile management”

The DFD LEVEL 0 of “FACULTY PROFILE MANAGEMENT” shown in fig 3.2.1. Here the visitors are faculty, administrator and users. Faculty Profile Maintenance is a system. Faculty can do edit, delete and view all the data and can do the filter view on Faculty Profile Maintenance and the users can view the faculty details.

3.2.2 DFD LEVEL 1

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

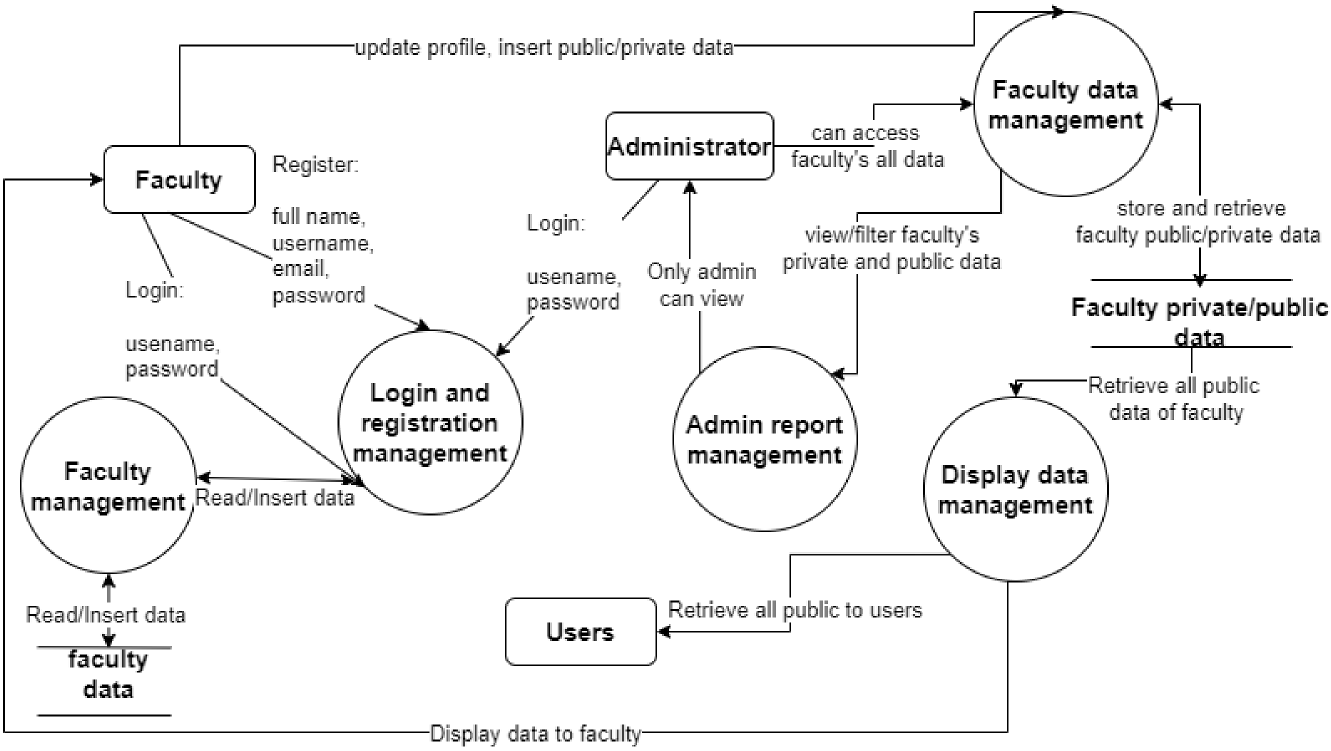


Fig 3.2.2 level 1 DFD for “faculty profile management”

The DFD LEVEL 1 of “FACULTY PROFILE MANAGEMENT” shown in fig 3.2.2. Here the Faculty Management, login & registration management, Admin report management, display data management & Faculty Data Management are systems. The visitors are faculty, users, administrator. Faculty Management can access the faculty database and the faculty data management can access the Faculty Private/Public database. The admin can also use the login & registration management system. The visitor faculty can access the faculty management through the login & registration management system where the data is stored in the database.

3.2.3 DFD LEVEL 2.1

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

Login and Registration Management:

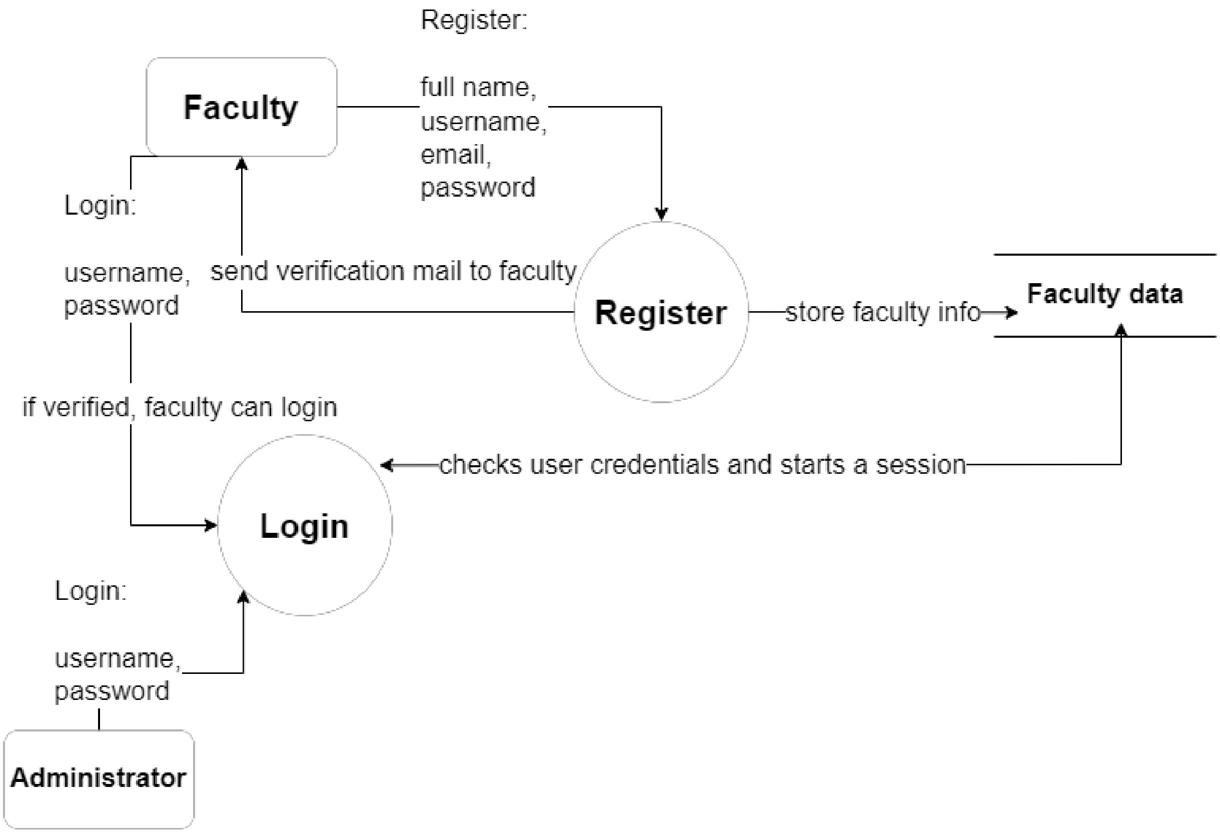


Fig 3.2.3 level 2.1 DFD for "faculty profile management"

The DFD LEVEL 2.1 of "FACULTY PROFILE MANAGEMENT" shown in fig 3.2.3. The system in the level 2 DFD are Register & Login and the visitors are Faculty and admin and the Database is Faculty data. The faculty can register through the register system by giving details of full name, user name, email &

password. The verification mail is sent to the faculty mail account and after verification they can login through the login system. The given details of the faculty members are stored in the faculty data. The admin can also use the required user id and password to login

3.2.4 DFD LEVEL 2.2

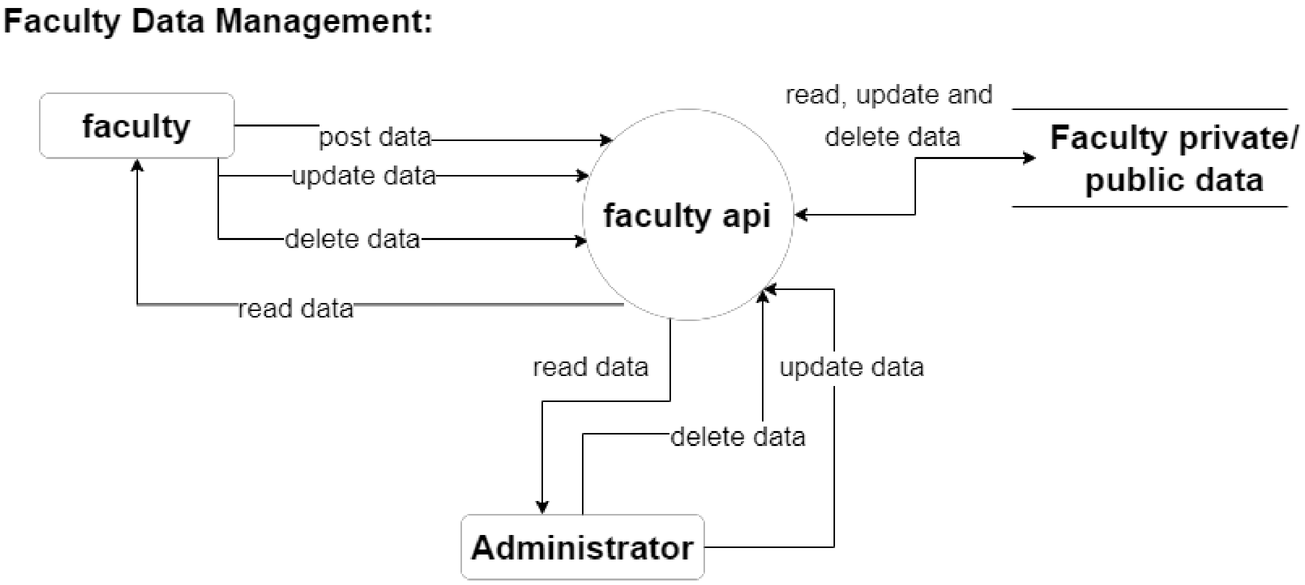


Fig 3.2.4 level 2.2 DFD for “faculty profile management”

The DFD LEVEL 2.2 of “FACULTY PROFILE MANAGEMENT” shown in fig 3.2.4. Here the visitors are 2 the database. The faculty can post, update, delete, read the data through faculty api system. The data will be stored in the faculty private/public database. The administrator can do read, delete, update the data through the faculty api system.

3.2.5 DFD LEVEL 2.3

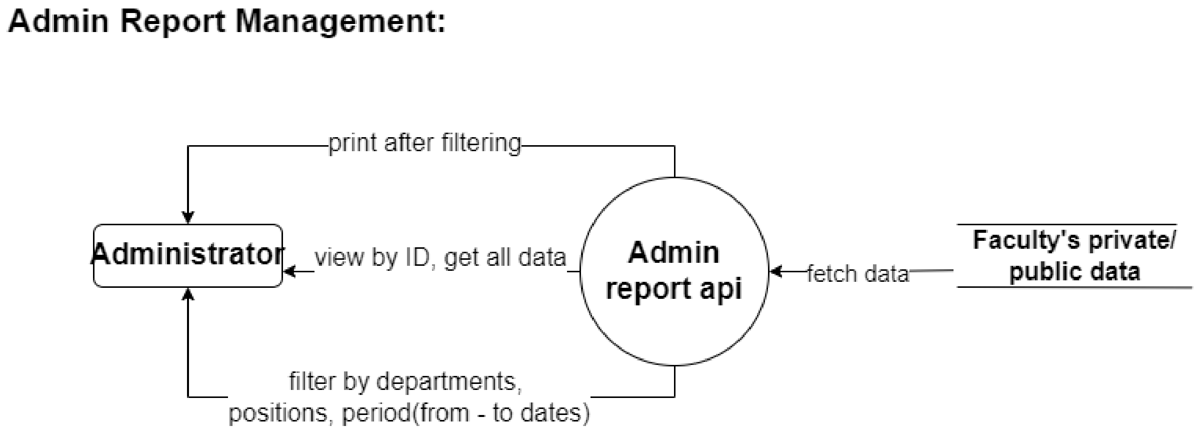


Fig 3.2.5 level 2.3 DFD for "faculty profile management"

The DFD LEVEL 2.3 of "FACULTY PROFILE MANAGEMENT" shown in fig 3.2.5. Here the visitors are administrator and the system is admin report api and the database is faculty private/public data. The administrator can generate report from the admin report api system. The admin can view the information through the faculty ID and can get all data of the faculty ID. The admin can filter the information by departments, position, period (from-to date). The admin can print the faculty details after filtering.

3.2.6 DFD LEVEL 2.4

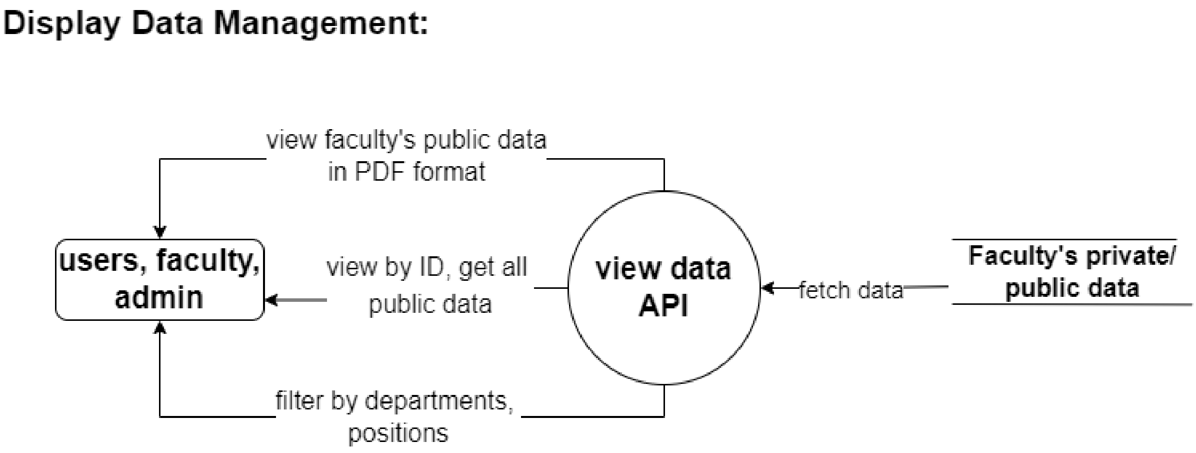

Display Data Management:
view faculty's public data in PDF format
users, faculty, admin
view by ID, get all public data
view data API
fetch data
Faculty's private/ public data
filter by departments, positions

Fig 3.2.6 level 2.4 DFD for “faculty profile management”

The DFD LEVEL 2.4 of “FACULTY PROFILE MANAGEMENT” shown in fig 3.2.6. Here the visitors are users. Faculty, admin and the system is view data API and the database is faculty private/public data. The administrator, users and faculty can view the faculty information through ID and can get public data through view data API system. The visitors can filter by departments and position of the faculty members. The visitors can view faculty’s public data in PDF format and can download the information’s in the form of PDF.

3.3 USECASE DIAGRAM

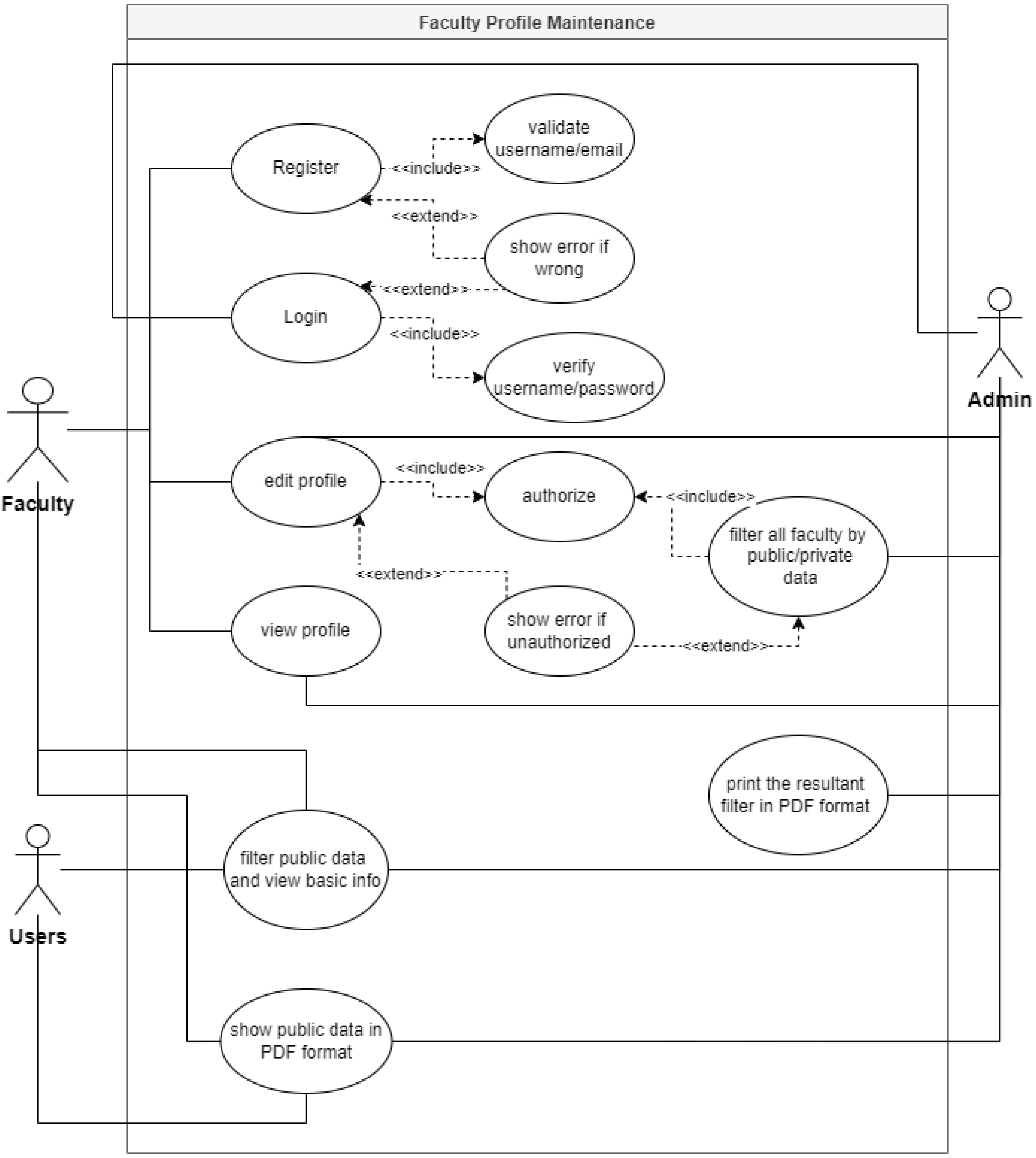


Fig 3.3 USECASE DIAGRAM

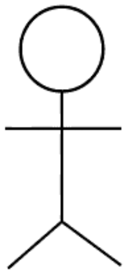
A use case diagram is a graphical depiction of a user's possible interactions with a system.

A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses

Purpose of Use Case Diagram

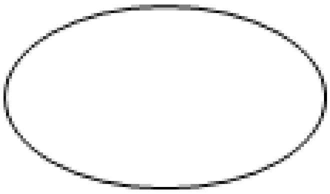
- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

Actor



An actor represents a role of a user that interacts with the system that you are modeling. The user can be a human user, an organization, a machine, or another external system.

USECASE

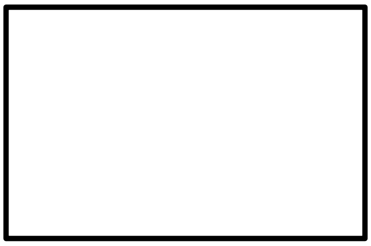


A use case describes a function that a system performs to achieve the user's goal. A use case must yield an observable result that is of value to the user of the system.

RELATIONSHIP

Relationship is a connection between model elements. A UML relationship is a type of model element that adds semantics to a model by defining the structure and behavior between the model elements.

SYSTEM BOUNDARY



The boundary, which defines the system of interest in relation to the world around it

Fig 3.3 represents the usecase diagram for faculty profile maintenance.

3.4 ARCHITECTURAL DIAGRAM

An architectural diagram is a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

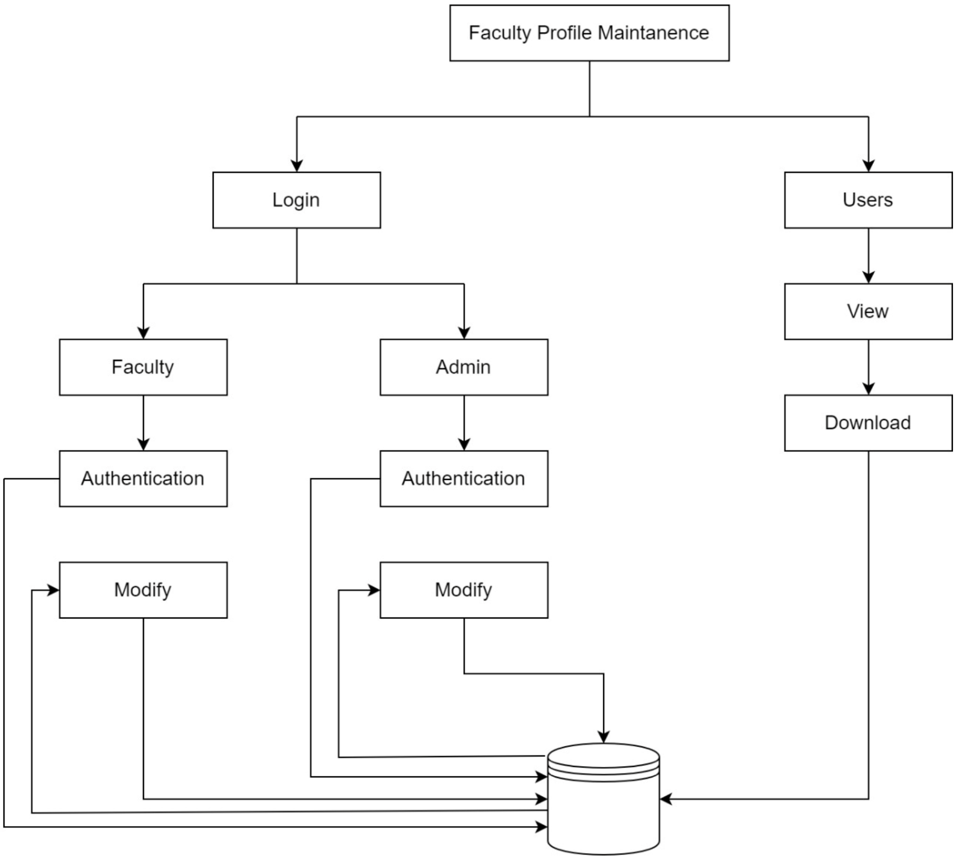
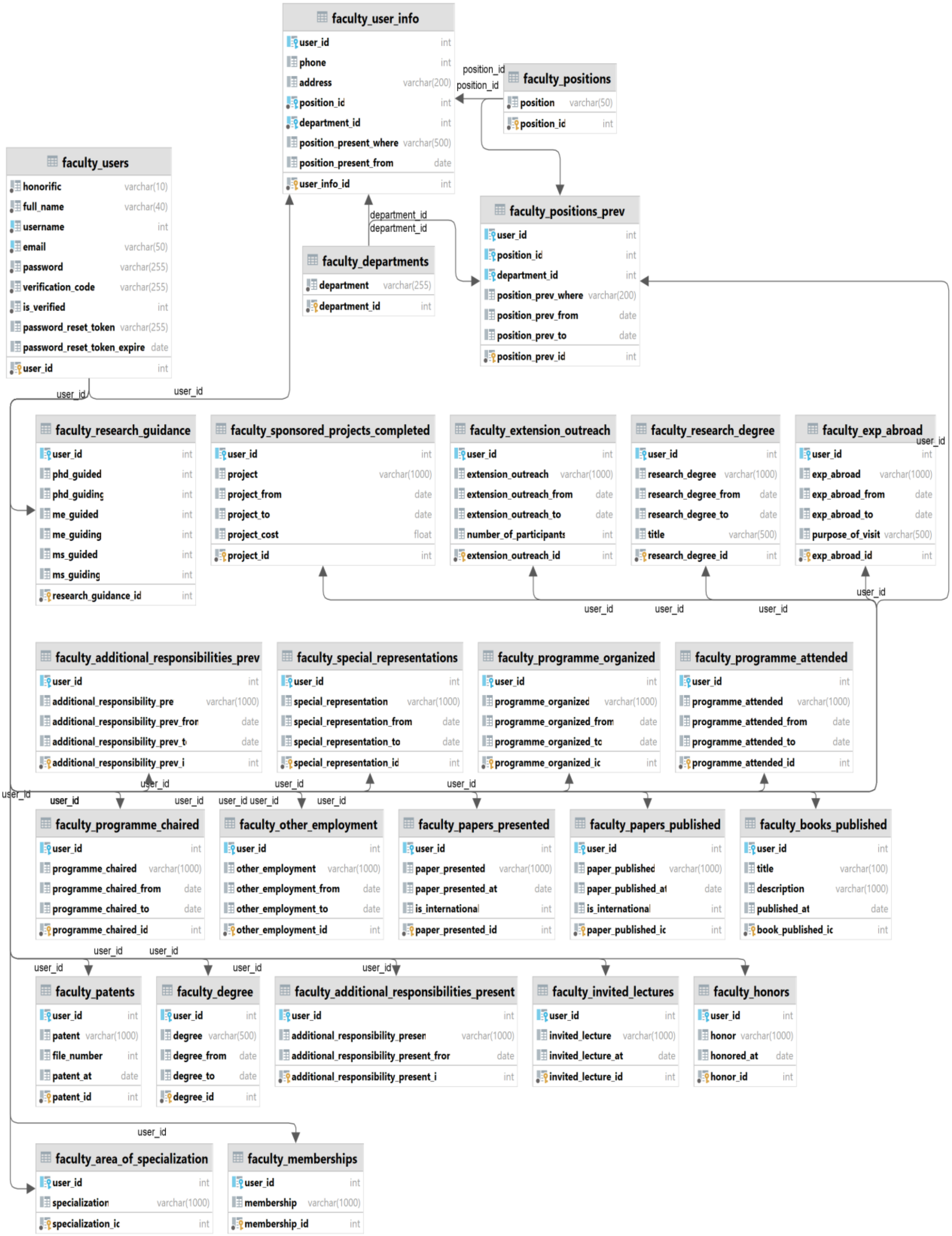


Fig 3.4 represents the architectural diagram for the faculty profile maintenance

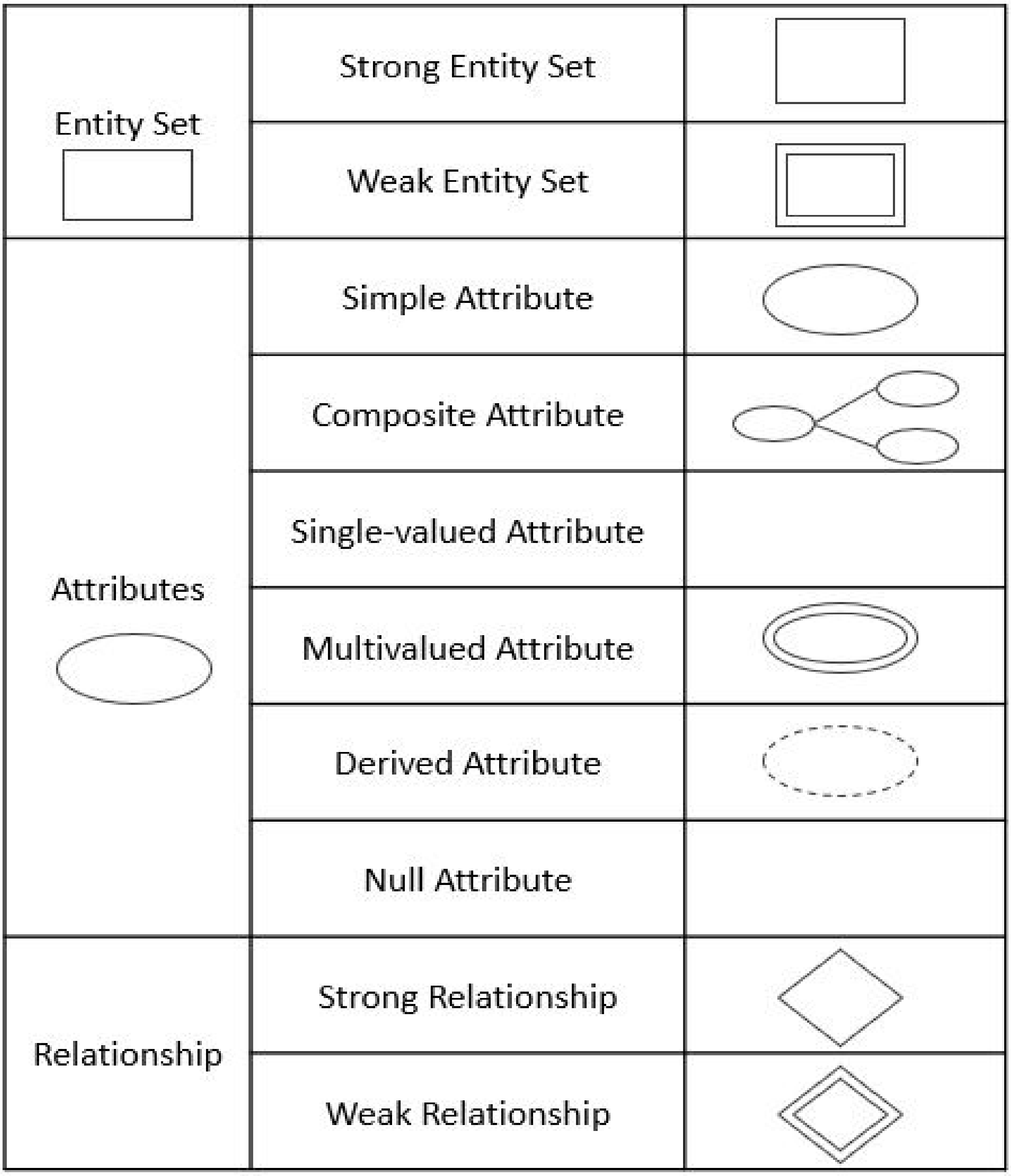
The Architecture Diagram is shown in Fig 3.4. In the Faculty Profile Maintenance system, the faculties and administrator should login to add and change the information of the faculty. The faculty can login by providing the faculty ID and password. The Authentication is verified by checking the information in the faculty profile maintenance database. Then they are allowed to change/edit the information. The edited information's are stored are also stored in the faculty profile maintenance database.

3.5 ENTITY RELATIONSHIP DIAGRAM



ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.

3.5.1 ER Notation

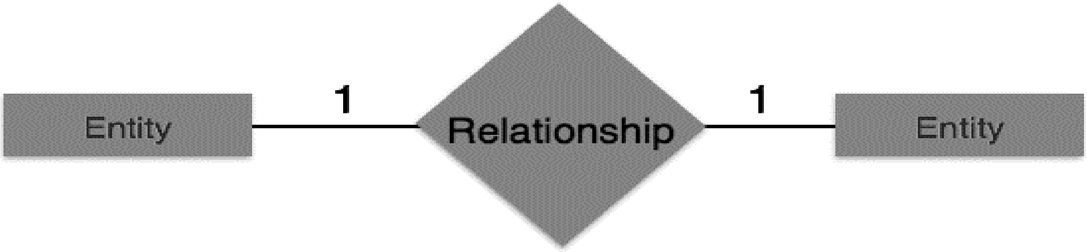


Entity Set	Strong Entity Set	
	Weak Entity Set	
Attributes	Simple Attribute	
	Composite Attribute	
	Single-valued Attribute	
	Multivalued Attribute	
	Derived Attribute	
	Null Attribute	
Relationship	Strong Relationship	
	Weak Relationship	

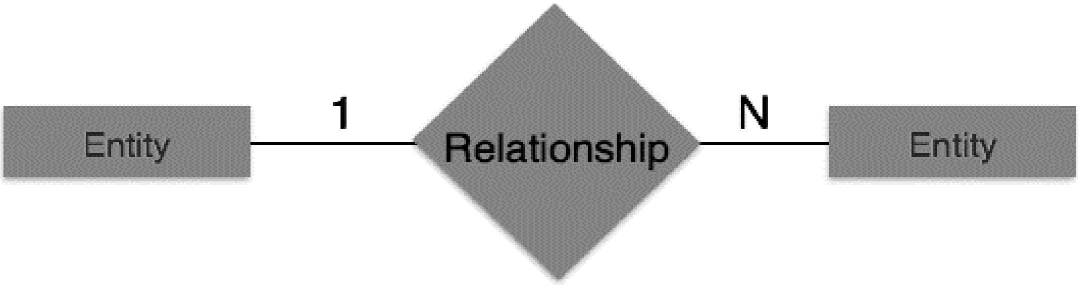
BINARY RELATIONSHIP AND CARDINALITY

A relationship where two entities are participating is called a binary relationship. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

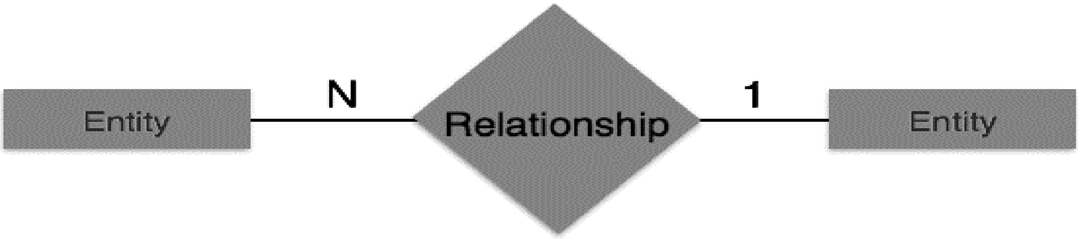
One-to-one − When only one instance of an entity is associated with the relationship, it is marked as '1:1'



One-to-many − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'

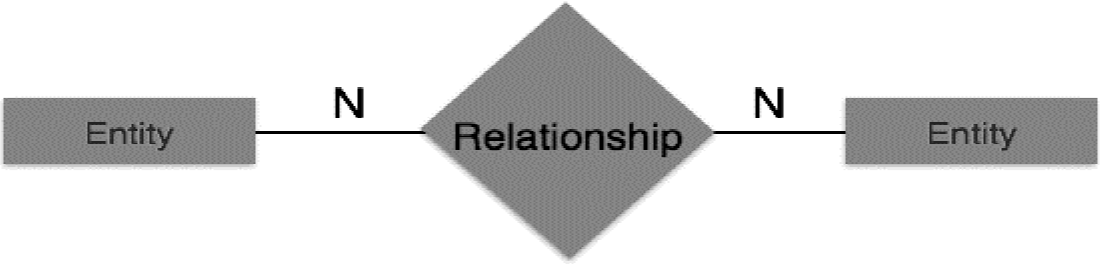


Many-to-one − When more than one instance of entity is associated with the



relationship, it is marked as 'N:1'

Many-to-many − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



CHAPTER – 4

SYSTEM DESCRIPTION

4.1 SYSTEM DEVELOPMENT

The system development process, also known as the system development life cycle, is a term used in the development of software where a set of methodical processes, activities, or phases are used to develop and implement a system

4.1.1 MODULES

A module is a collection of source files and build settings that allow you to divide your project into discrete units of functionality.

Module is a logical separation of a functionality within a project.

We can have as many modules in an application, they are basically used for reusability and better code maintenance

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. An enterprise-level software application may contain several different modules, and each module serves unique and separate business operations.

Modules make a programmer's job easy by allowing the programmer to focus on only one area of the functionality of the software application. Modules are typically incorporated into the program (software) through interfaces.

There are two modules in this project.

4.1.2 MODULAR DESCRIPTION

The modules in the system are

1. AUTHENTICATION
2. FACULTY
    a. PUBLIC DATA
    b. PRIVATE DATA
3. ADMIN
4. VIEW PROFILE

4.1.2.1 AUTHENTICATION:

The user can register to Faculty Profile Maintenance system by visiting the registration page. The user should provide some basic information about them and the Faculty ID, E-mail ID and the password should be entered.

After successfully registering in the system, the user will receive a verification mail for the registered E-mail ID. The verification mail contains a verification link, user can verify their profile by clicking the link.

Admin can also use the same registration page for registration, but some credentials should be altered in the DB, to show that this person is the Admin. The credentials are present in the faculty_users table, credentials to change is_admin and is_verified from 0 to 1.

Users can login by using the provided Faculty ID/E-mail ID and password, after successfully logged in, a session will be created for the user, and this session contains some basic info about the user who logged in.

User can also use the ‘Forget Password?’ functionality provided in this this system to later change their password. After clicking the ‘Forget Password?” button the user should provide an E-mail ID. If the E-mail ID is a registered and valid E-mail ID, the user will receive a mail which contains an URL to change password. User can visit the URL to enter new password.

‘The verification link and forget password link sent to the mail are only valid for one day.’

4.1.2.2 FACULTY:

Faculty as a user to this system can Insert/Update/Delete their public/private information. Only verified user can Insert/Update/Delete data.

4.1.2.2.1 PUBLIC DATA:

After successfully logged in to the profile, the user will be redirected to a page where the user can edit their basic info and upload their profile picture. If the user already entered information before, after the revisiting the page, the information will still present same input boxes, where they can update their information or can delete it. If an unverified user entered their information and clicked the submit/update button, a warning will be displayed as ‘User not verified’. Unless the user’s basis info other information such as Programme Organized, the user can enter a list of data along with the month and year they have organized the programme. Public data is included in the profile PDF.

4.1.2.2.2 PRIVATE DATA

The user can enter their private data such as on duty orders, Invigilation duties in the private information section. User’s public data is visible to all people, but private data is only visible to them and the Admin. Private data is not included in the profile PDF.

4.1.2.3 ADMIN:

Admin can use the basic registration page used by other users for registration. But some credentials should be altered in the DB, to show that this person is the Admin. The credentials are present in the faculty_users table, credentials to change is_admin and is_verified from 0 to 1.

Admin can enter into the profile section by means of entering their Admin ID and password. Once the Admin gets logged in, Admin can view the faculty profile information. The Admin can make the report of the faculty member's both private and public data (or) information at a particular period of time. The Admin can also view their information and can download the faculty profile pdf anytime.

In the report section the following information are provided: Category, From Date, To Date, and Department (optional). After giving the required information, the data will be displayed in a format. The admin can generate a report in PDF format or can view the faculty details by directly visiting their profile.

4.1.2.4 VIEW PROFILE:

In the Home page, any user or other people who visit the webpage can view the faculty Profile by Department. Only the public data is visible to them in a separate page. By clicking the Generate PDF button, the visitors can download the faculty's public profile in a PDF format.

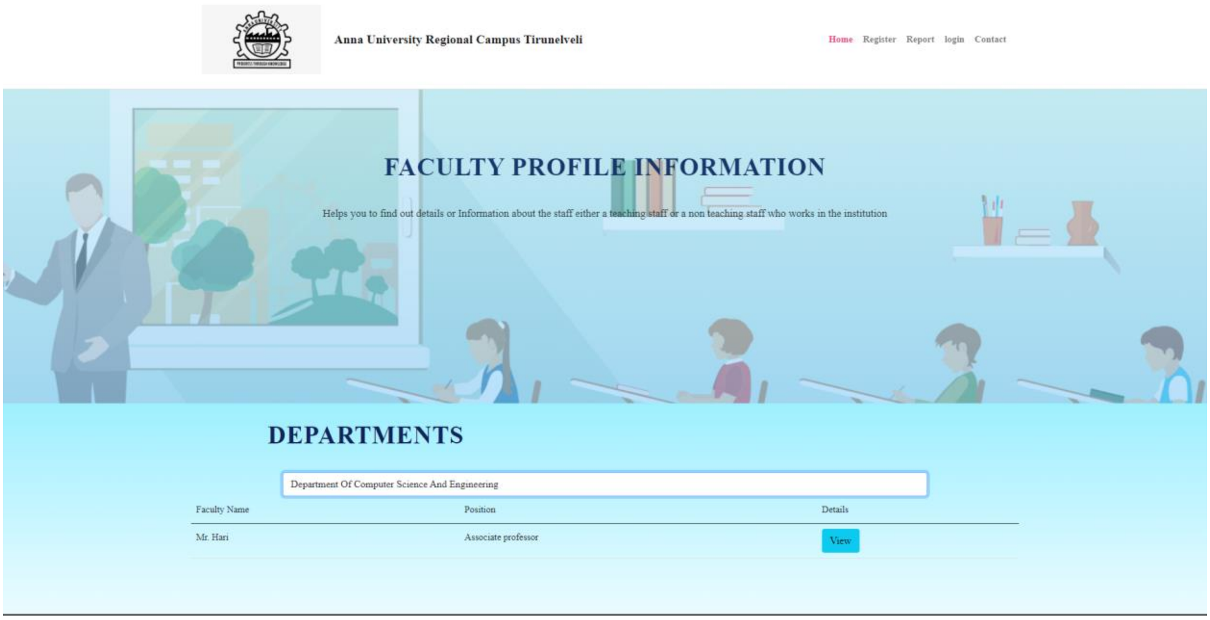
CHAPTER 5

SYSTEM IMPLEMENTATION AND TESTING

5.1 SYSTEM IMPLEMENTATION

Systems implementation is the process of: defining how the information system should be built (i.e., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standard (i.e., quality assurance).

Faculty Profile Maintenance Home page:



Anna University Regional Campus Tirunelveli

FACULTY PROFILE INFORMATION

Helps you to find out details or Information about the staff either a teaching staff or a non teaching staff who works in the institution

DEPARTMENTS

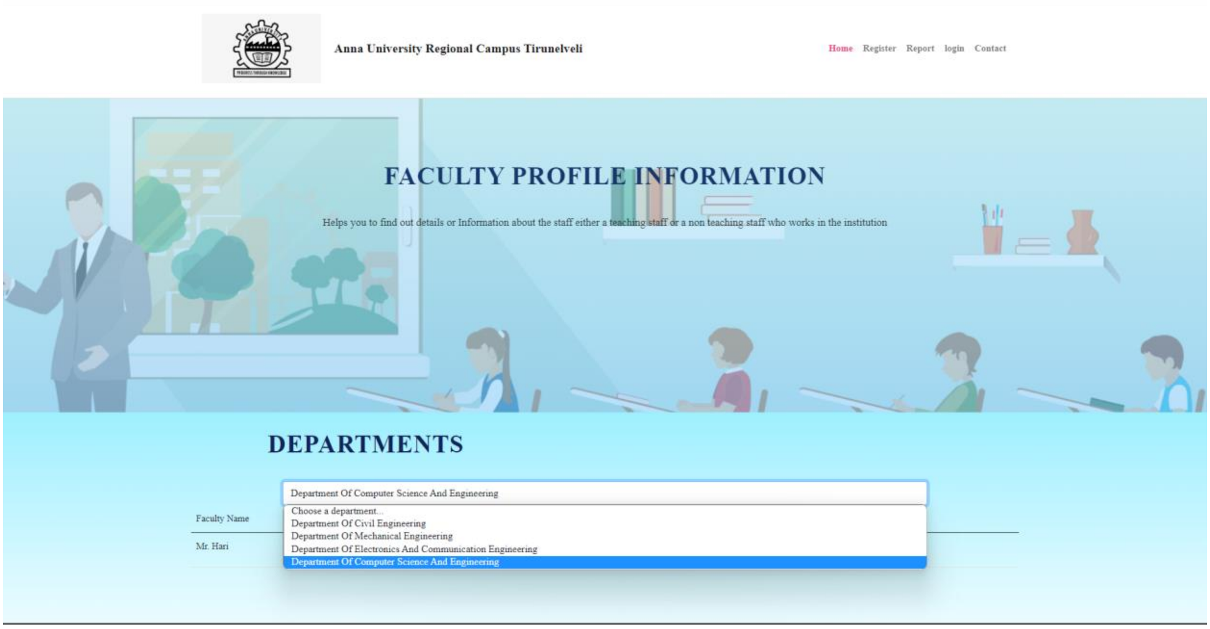
Department Of Computer Science And Engineering

Faculty Name	Position	Details
Mr. Hari	Associate professor	View

Fig 5.1.1 Faculty Profile Maintenance Home page

The Home page of "FACULTY PROFILE MAINTENANCE" shown in fig 5.1.1

Faculty Profile Maintenance Home page:



Anna University Regional Campus Tirunelveli

FACULTY PROFILE INFORMATION

Helps you to find out details or Information about the staff either a teaching staff or a non teaching staff who works in the institution

DEPARTMENTS

Department Of Computer Science And Engineering

Choose a department...
Department Of Civil Engineering
Department Of Mechanical Engineering
Department Of Electronics And Communication Engineering
Department Of Computer Science And Engineering

Faculty Name

Mr. Hari

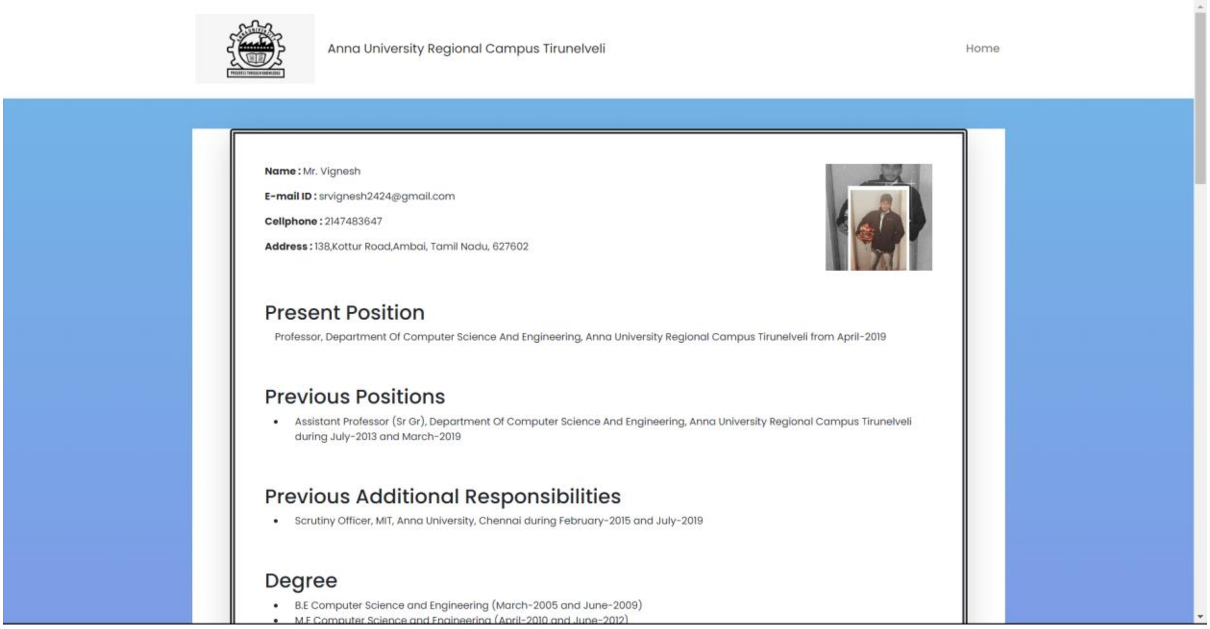
Fig 5.1.2 Faculty Profile Maintenance Home page

The home page contains all the navigation link to other pages such as register, faculty login and admin login. In the home page there will be a department section. you can select any department in the department section. After you selected the list of all the faculty members will be shown down in the drop-down section. A list of faculty members name will be shown below and if you click view on a particular faculty member. This page will redirect to another page which contain all the information in a form of pdf

The department sections the following options

- Department of Civil Engineering
- Department of Mechanical Engineering
- Department of Electronics and Communication Engineering
- Department of Computer Science and Engineering

Faculty Profile Maintenance view profile page:



Name : Mr. Vignesh

E-mail ID : srvignesh2424@gmail.com

Cellphone : 2147483647

Address : 138,Kottur Road,Ambai, Tamil Nadu, 627602

Present Position

Professor, Department Of Computer Science And Engineering, Anna University Regional Campus Tirunelveli from April-2019

Previous Positions

- Assistant Professor (Sr Gr), Department Of Computer Science And Engineering, Anna University Regional Campus Tirunelveli during July-2013 and March-2019

Previous Additional Responsibilities

- Scrutiny Officer, MIT, Anna University, Chennai during February-2015 and July-2019

Degree

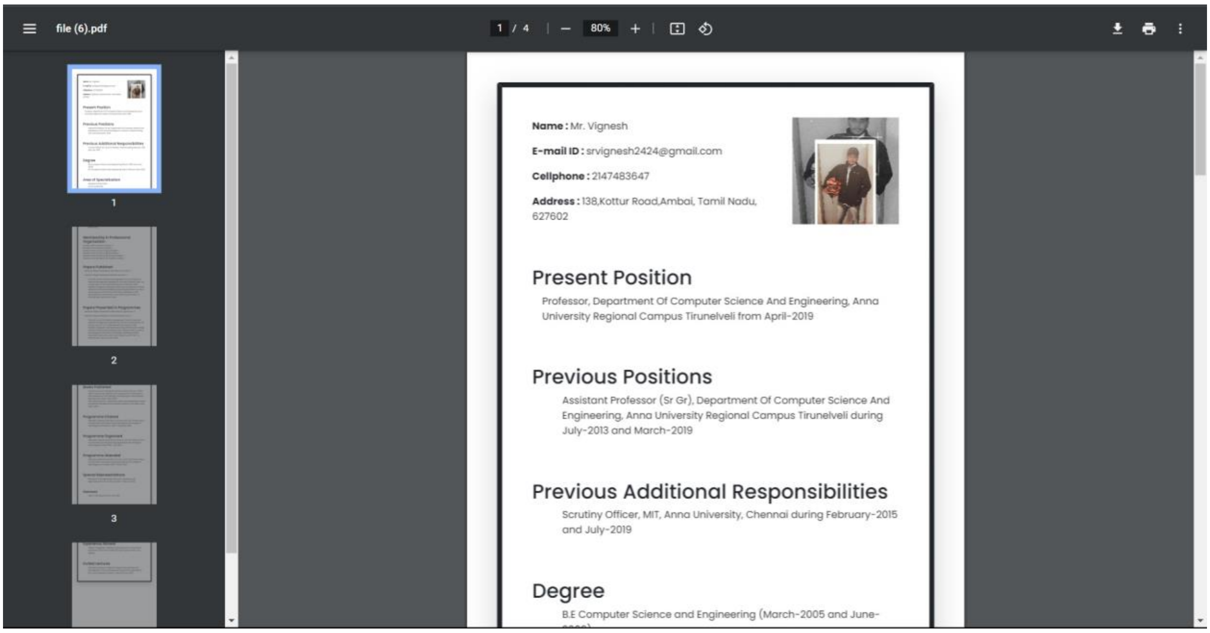
- B.E Computer Science and Engineering (March-2005 and June-2009)
- M.E Computer Science and Engineering (April-2010 and June-2012)

Fig 5.1.3 Faculty Profile Maintenance view profile page

The view profile page of "FACULTY PROFILE MAINTENENCE" shown in fig 5.1.3

In this page all the visitors of the website can view any profile of any department of the faculty members.in this page all the public data of the faculty members are shown and the private data cannot be viewed by the visitors of the website. In this page there will be an option called generate pdf by clicking the option the user or the visitor can download the details of the faculty as a form of pdf.

Faculty Profile Maintenance view profile pdf page:



Name : Mr. Vignesh

E-mail ID : srvignesh2424@gmail.com

Cellphone : 2147483647

Address : 138,Kottur Road,Ambai, Tamil Nadu, 627602

Present Position

Professor, Department Of Computer Science And Engineering, Anna University Regional Campus Tirunelveli from April-2019

Previous Positions

Assistant Professor (Sr Gr), Department Of Computer Science And Engineering, Anna University Regional Campus Tirunelveli during July-2013 and March-2019

Previous Additional Responsibilities

Scrutiny Officer, MIT, Anna University, Chennai during February-2015 and July-2019

Degree

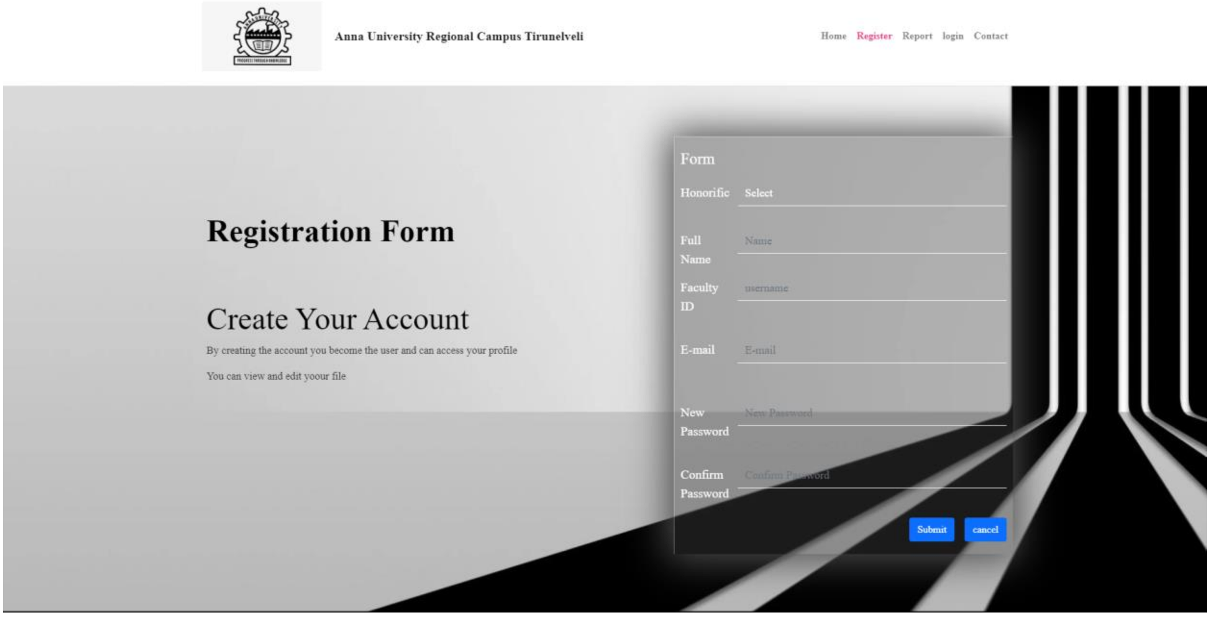
B.E Computer Science and Engineering (March-2005 and June-

Fig 5.1.4 Faculty Profile Maintenance view profile pdf page

The view profile page of “FACULTY PROFILE MAINTENENCE” shown in fig 5.1.4

In this page the downloaded pdf will be viewed

Faculty Profile Maintenance Registration page:



Registration Form

Create Your Account

By creating the account you become the user and can access your profile

You can view and edit yoour file

Form

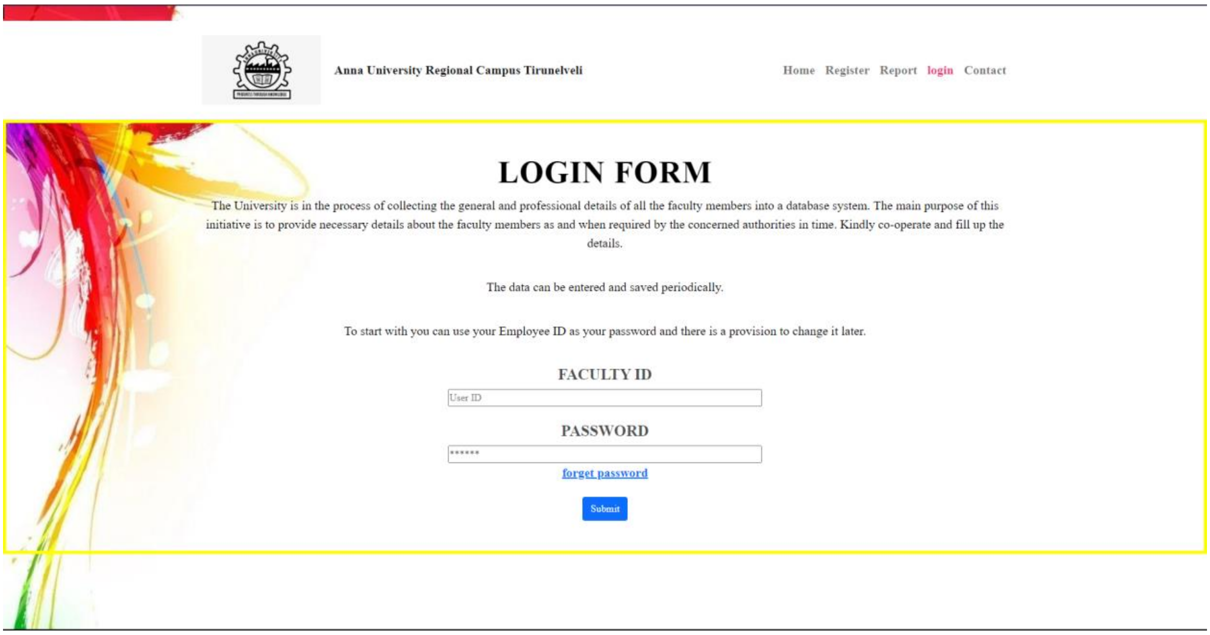
Honorific	Select
Full Name	Name
Faculty ID	username
E-mail	E-mail
New Password	New Password
Confirm Password	Confirm Password

Fig 5.1.5 Faculty Profile Maintenance Registration page

The Registration page of “FACULTY PROFILE MAINTENENCE” shown in fig 5.1.5

In this registration page the existing faculty member or a new faculty member can be registered as a new user. They can register as a new user by giving certain details like email id and can create a password for themselves. After they enter the details given in the registration form, they can click the option submit and an verify mail will be sent to their mail account and after they verify the can login. With the given faculty id and the password, they can login to their account and edit their information

Faculty Profile Maintenance Login page:



LOGIN FORM

The University is in the process of collecting the general and professional details of all the faculty members into a database system. The main purpose of this initiative is to provide necessary details about the faculty members as and when required by the concerned authorities in time. Kindly co-operate and fill up the details.

The data can be entered and saved periodically.

To start with you can use your Employee ID as your password and there is a provision to change it later.

FACULTY ID

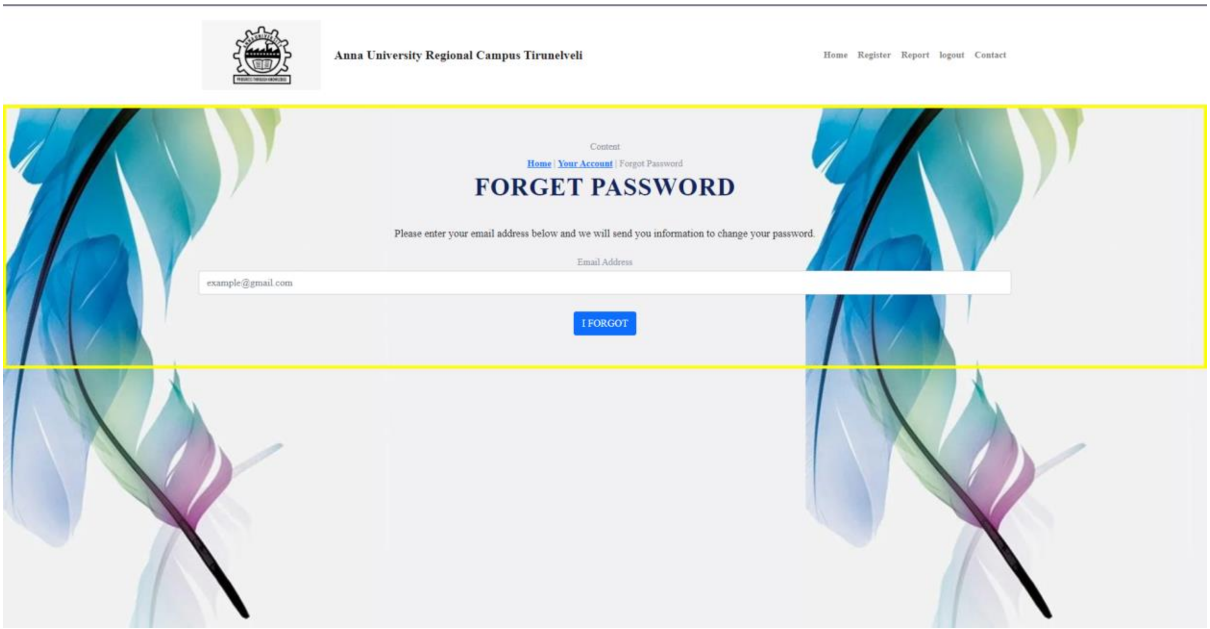
PASSWORD

Fig 5.1.6 Faculty Profile Maintenance Login page

The Login page of “FACULTY PROFILE MAINTENENCE” shown in fig 5.1.6.

The login page can only be accessed by the faculty and the admin. The faculty and the admin should need a faculty id to login. Without the faculty id the cannot login. Faculty id and password are required to get logged in to this page. If the user does not remember the password or the faculty id, they can choose the option forgot password and it will redirect to a new page.

Faculty Profile Maintenance Forgot password page:


Anna University Regional Campus Tirunelveli
Home Register Report logout Contact
Content
Home | Your Account | Forgot Password
I FORGOT

FORGET PASSWORD

Please enter your email address below and we will send you information to change your password.

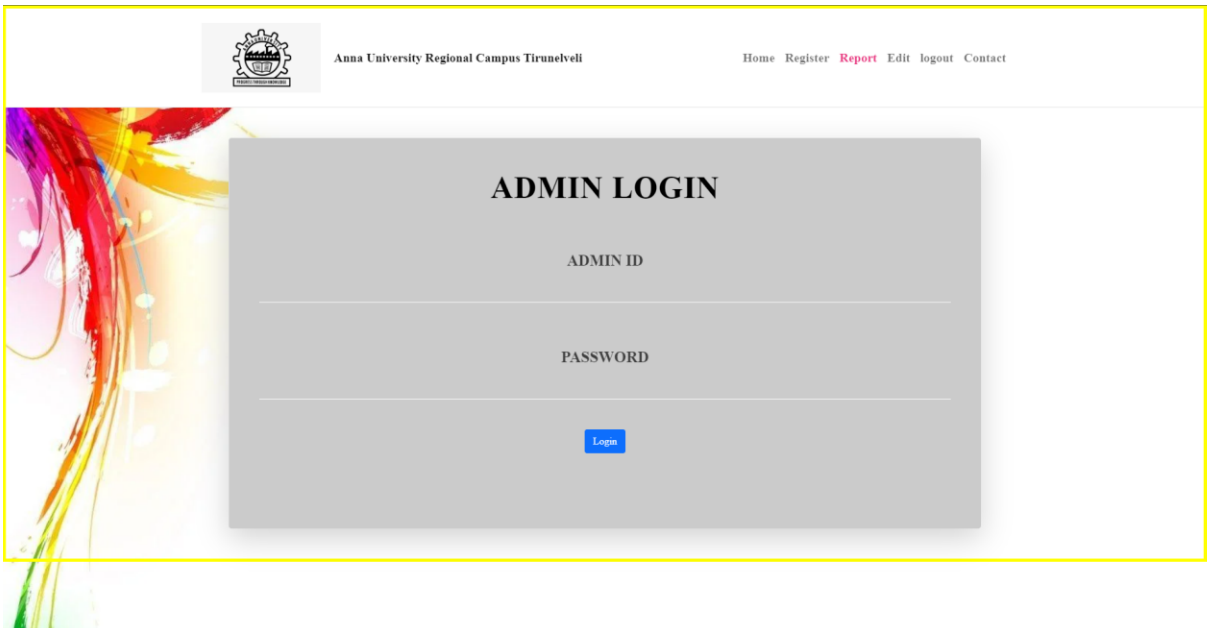
Email Address: example@gmail.com

Fig 5.1.7 Faculty Profile Maintenance Forgot password page

The forgot password page of "FACULTY PROFILE MAINTENANCE" shown in fig 5.1.7

If the user or the faculty member does not remember the given mail id or the password, they can click the forgot password button in the login page and it will redirect to the forgot password page. in this page the user gives a mail id and click on I forgot a mail will be sent to their mail account and from there they can change the password and can login as usual

Faculty Profile Maintenance Admin page:



Anna University Regional Campus Tirunelveli

ADMIN LOGIN

ADMIN ID

PASSWORD

Fig 5.1.8 Faculty Profile Maintenance Admin page

The admin login page of “FACULTY PROFILE MAINTENANCE” shown in fig 5.1.8

The admin can also register as a new user or the admin will have a unique id and unique password to login. This page is used to login only for the admin and can only accessed by the admin. the admin should give required id and password to login.

Faculty Profile Maintenance Report page:

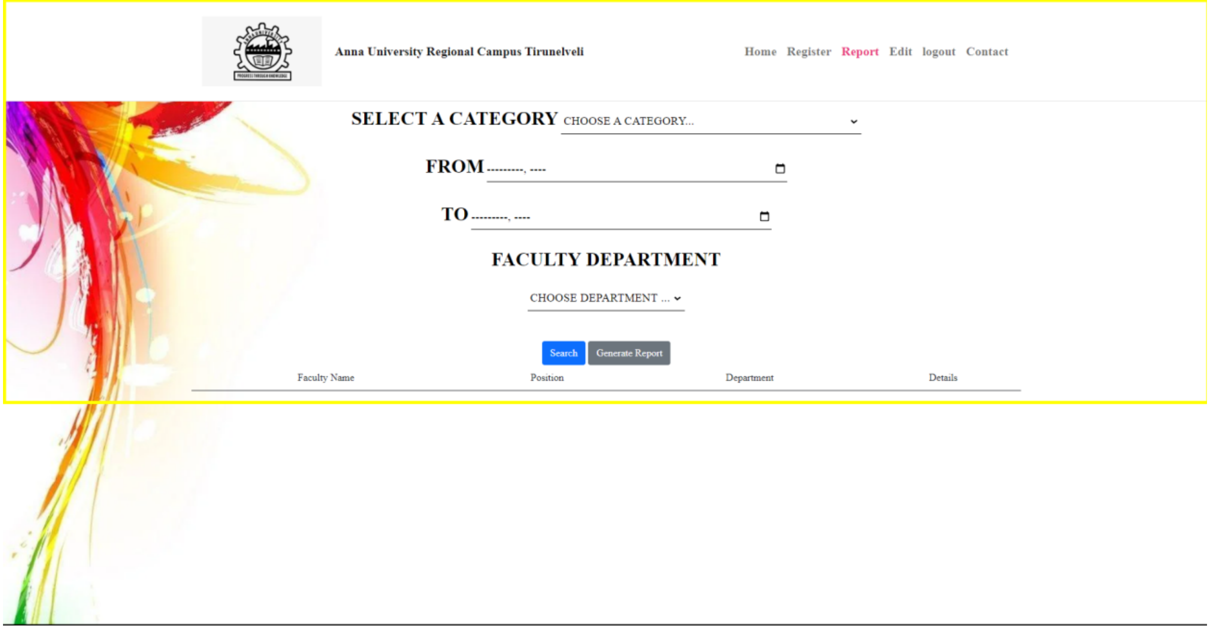
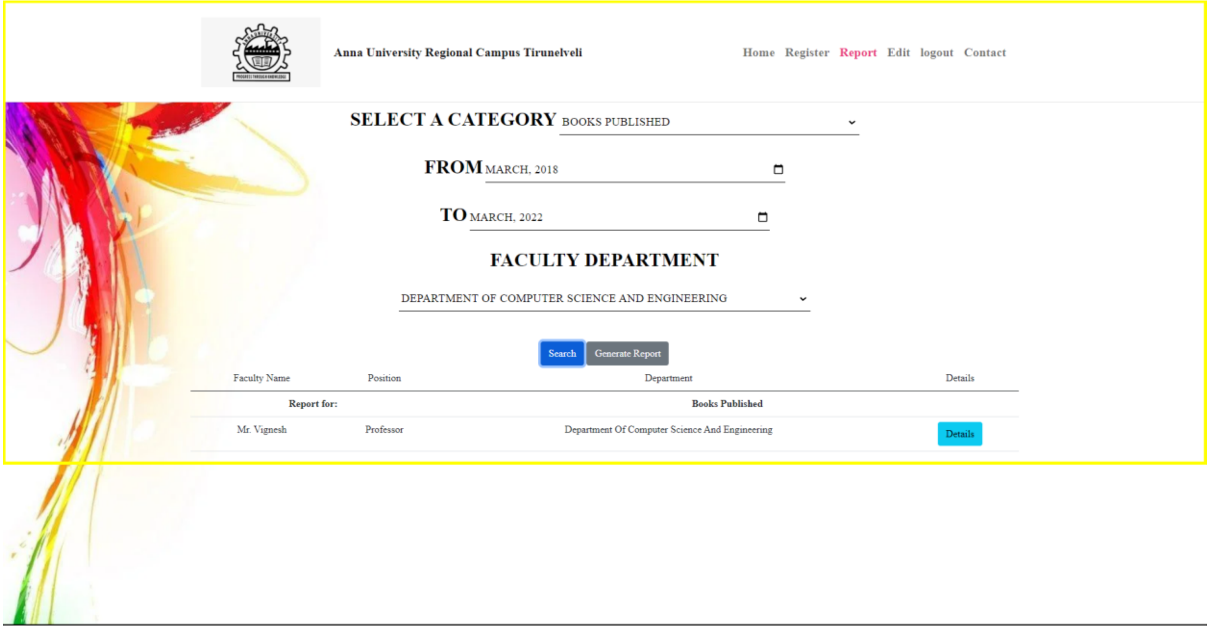


Fig 5.1.9 Faculty Profile Maintenance Admin page

The admin login page of “FACULTY PROFILE MAINTENANCE” shown in fig 5.1.9

The admin can generate report with this form. The admin should select a category or fields in the select a category option. They can choose the date from and to and the admin select a department and click search the results will be shown below.in the results there will be a view button. if admin clicks the view button this page will redirects to the next page.

Faculty Profile Maintenance Generating Report page:



SELECT A CATEGORY BOOKS PUBLISHED

FROM MARCH, 2018

TO MARCH, 2022

FACULTY DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

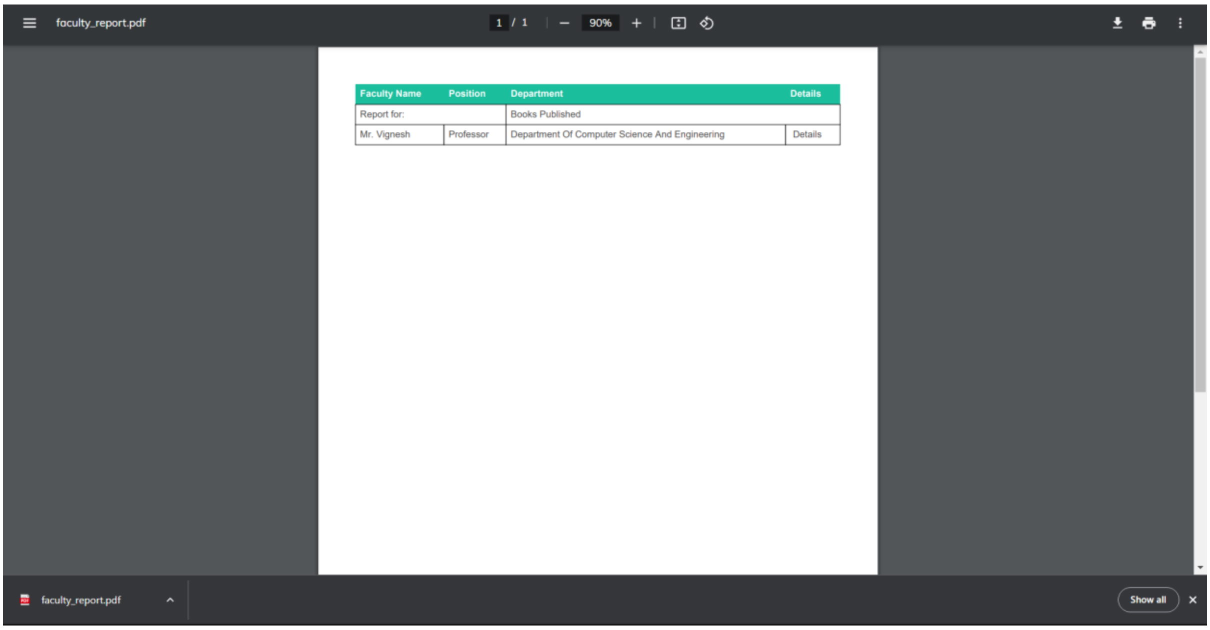
Faculty Name	Position	Department	Details
Report for:		**Books Published**	
Mr. Vignesh	Professor	Department Of Computer Science And Engineering	Details

Fig 5.1.10 Faculty Profile Maintenance generating report page

The generating report page for admin of "FACULTY PROFILE MAINTENANCE" shown in fig 5.1.10

By selecting the options in this page. After clicking the search button, a list of details will be displayed below.by selecting the details button the page will redirects to the edit page which contains the information of the particular faculty. By selecting the generate report the details will be downloaded in the form of pdf

Faculty Profile Maintenance Pdf Report page:

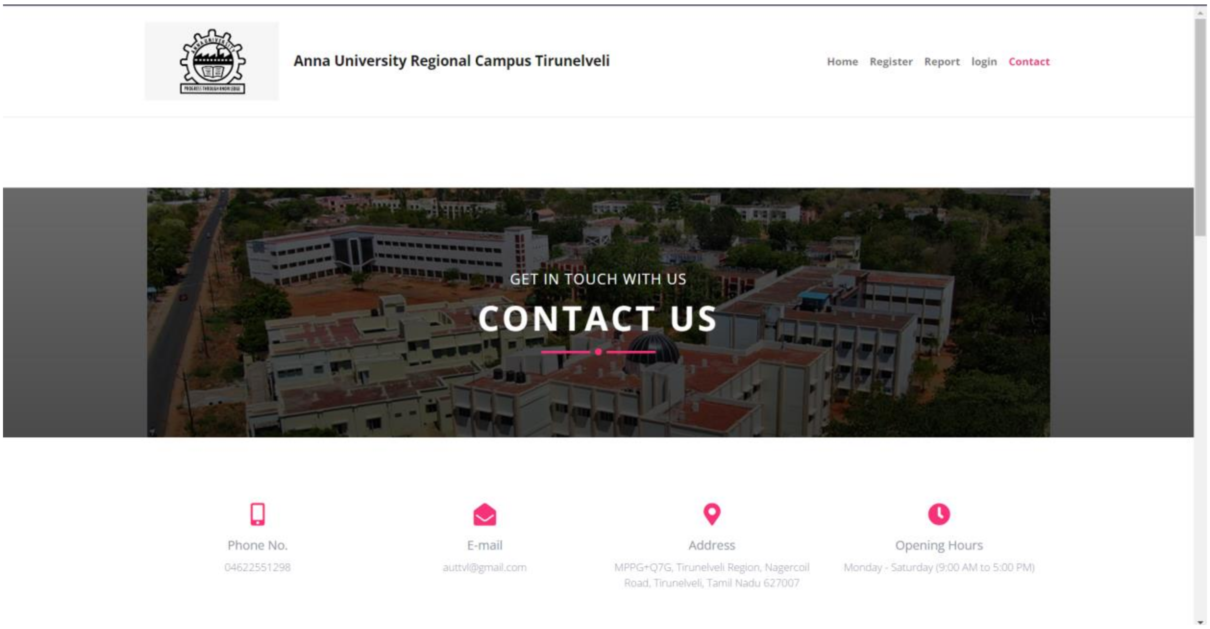


Faculty Name	Position	Department	Details
Report for:		Books Published	
Mr. Vignesh	Professor	Department Of Computer Science And Engineering	Details

Fig 5.1.11 Faculty Profile Maintenance report pdf page

The report will be downloaded in the form of pdf of "FACULTY PROFILE MAINTENANCE" shown in fig 5.1.11.

Faculty Profile Maintenance Contact page:



Anna University Regional Campus Tirunelveli

GET IN TOUCH WITH US

CONTACT US

Phone No.
04622551298

E-mail
auttvl@gmail.com

Address
MPPG+Q7G, Tirunelveli Region, Nagercoil Road, Tirunelveli, Tamil Nadu 627007

Opening Hours
Monday - Saturday (9:00 AM to 5:00 PM)

Fig 5.1.12 Faculty Profile Maintenance Contact page

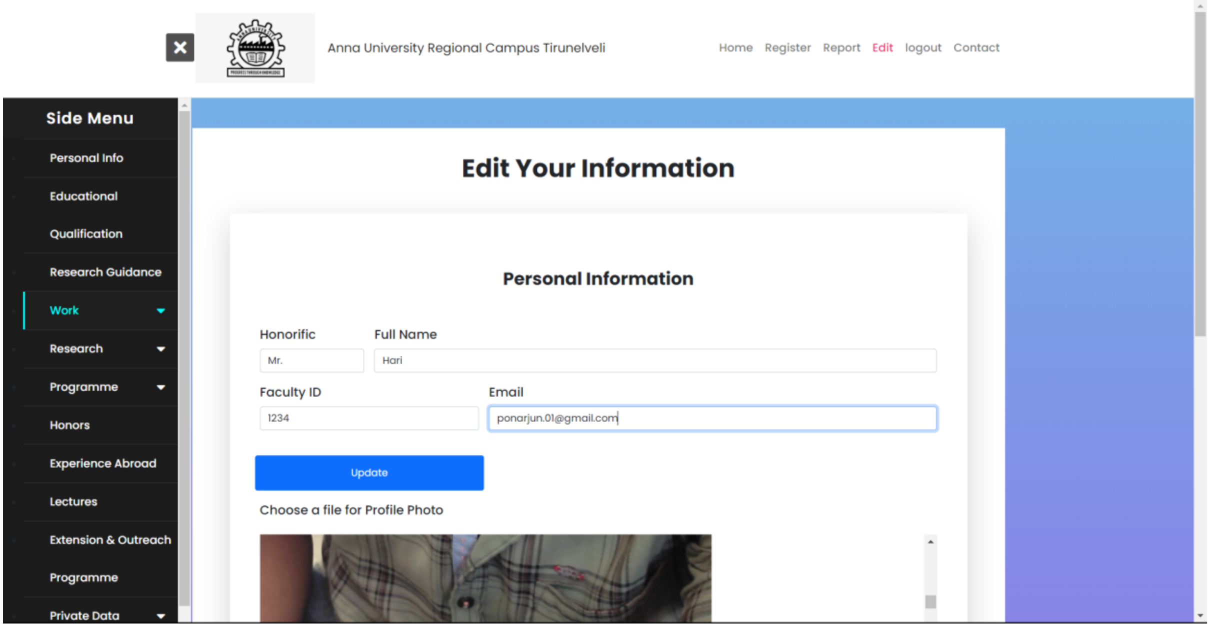
The contact page of “FACULTY PROFILE MAINTENANCE” shown in fig 5.1.12.

Their contact page **includes all the standard information — name, location, email, and so forth**. The page feels plain, but it fits with the rest of their site and their overall branding, which lets the user know who the brand is and what they can expect.

Our website contact page contains the following basics:

- Phone number of the institution
- Addresses of the institution
- Email addresses of the institution
- Opening hours of the institution
- Geolocation maps
- Some media links

Faculty Profile Maintenance Edit information page:



Edit Your Information

Personal Information

Honorific: Mr.

Full Name: Hari

Faculty ID: 1234

Email: ponarjun.01@gmail.com

Choose a file for Profile Photo

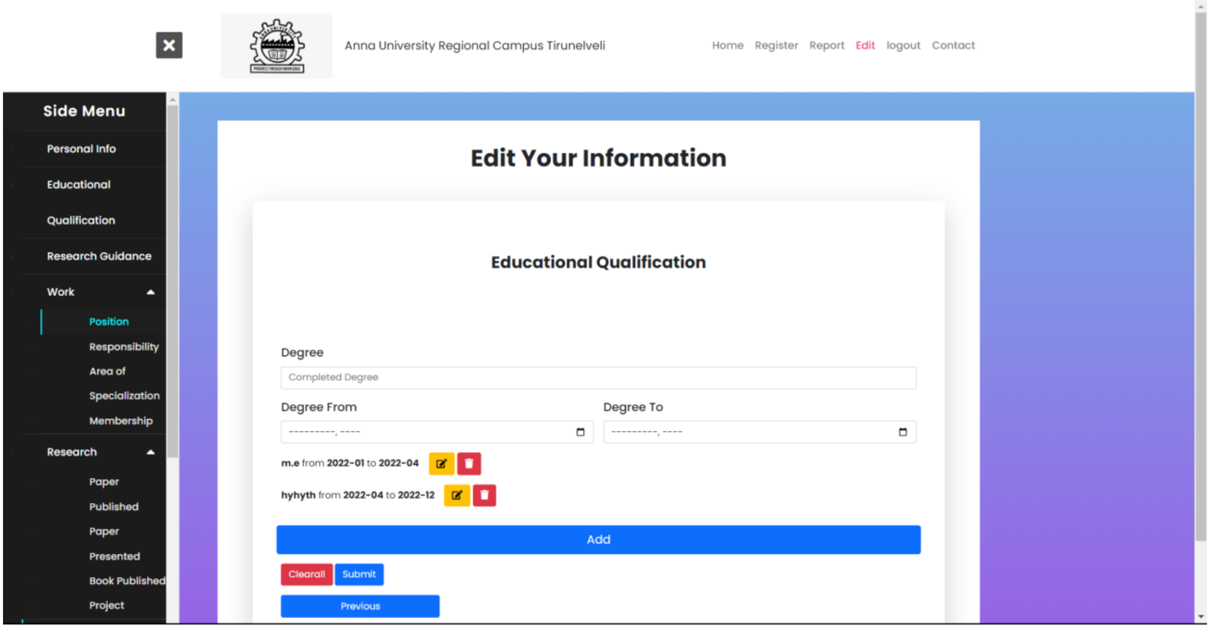
Fig 5.1.13 Faculty Profile Maintenance edit information page

The contact page of “FACULTY PROFILE MAINTENANCE” shown in fig 5.1.13.

After the login the faculty members can add their information and if they need any changes they can modify or edit their information anytime. If they want to remove the added information, they can remove their information by the delete option. It also provides the functionality of downloading information’s or forms in a form of pdf for printing purpose. In this page the user can add files such as image or pdf for their information

In this page the user needs to give their personal information.

Faculty Profile Maintenance Public data:



Anna University Regional Campus Tirunelveli

Edit Your Information

Educational Qualification

Degree: Completed Degree

Degree From: ---------, ----

Degree To: ---------, ----

m.e from 2022-01 to 2022-04

hyhyth from 2022-04 to 2022-12

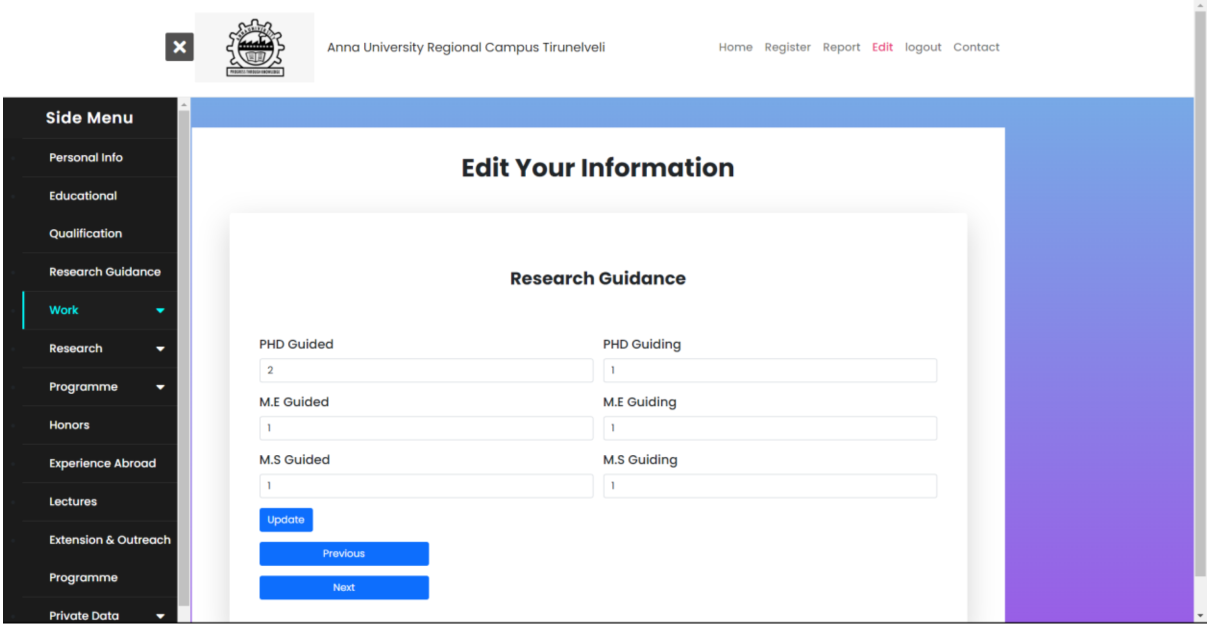
Fig 5.1.14 Faculty Profile Maintenance public data

The edit page information's can be classified as two types such as the information contains public data and the private data

The fig 5.1.14 show the information that contains public data of educational qualification of the faculty, the public data can be viewed by anyone with no restriction.

This page gets the information about the faculty's educational qualification. In this page the user needs to give the details of what degree they studied and from when to when they studied are the details the user needs to give.

Faculty Profile Maintenance Research Guidance form:



Edit Your Information

Research Guidance

PHD Guided: 2
PHD Guiding: 1
M.E Guided: 1
M.E Guiding: 1
M.S Guided: 1
M.S Guiding: 1

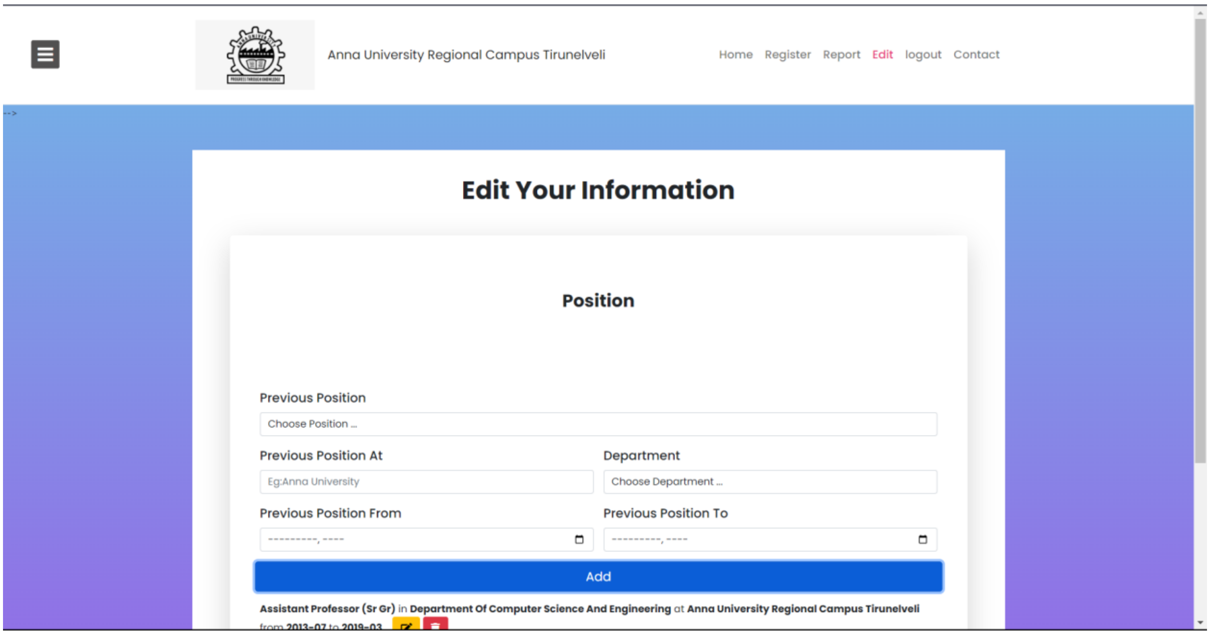
Fig 5.1.15 Faculty Profile Maintenance research guidance data

The research guidance page of the faculty profile maintenance will contain the public data.

The fig 5.1.15 show the information that contains public data for research guidance of the faculty, the public data can be viewed by anyone with no restriction.

In this page the user or the faculty needs to give the details of PhD guided and guiding and M.E guided and guiding and should press the update button and the data will be stored in the database If they need to move to the next page, they can click next button and it will redirect to the next page.

Faculty Profile Maintenance Position form:



Edit Your Information

Position

Previous Position

Choose Position ...

Previous Position At: Eg:Anna University

Department: Choose Department ...

Previous Position From: ---------, ----

Previous Position To: ---------, ----

Assistant Professor (Sr Gr) in **Department Of Computer Science And Engineering** at **Anna University Regional Campus Tirunelveli** from **2013-07** to **2019-03**

Fig 5.1.16 Faculty Profile Maintenance private data

The position form page of the faculty profile maintenance will contain the public data.

The fig 5.1.16 show the information that contains public data for position of the faculty member who works in the institution, the public data can be viewed by anyone with no restriction.

In this page the user or the faculty needs to give the details about their previous positions of the work the done in the institution and they can add multiple information by clicking the add button by clicking the add button the data will be stored in the database and it will be viewed below if they need any changes the edit or delete the information. If they need to move to the next page, they can click next button and it will redirect to the next page.

Faculty Profile Maintenance Private data:

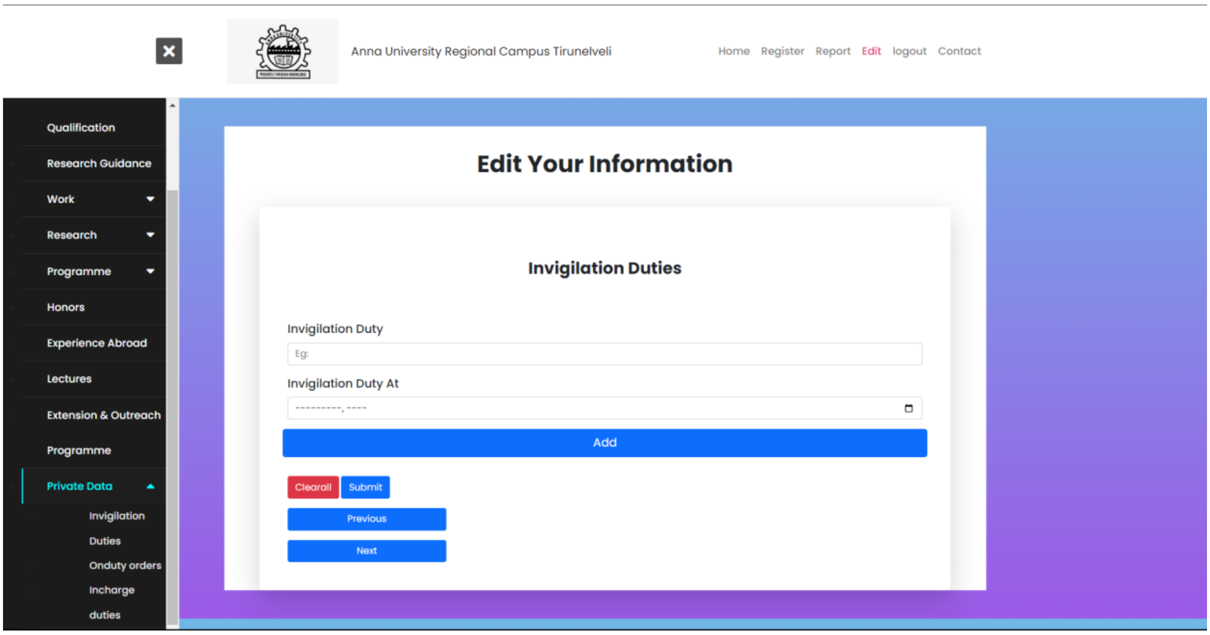
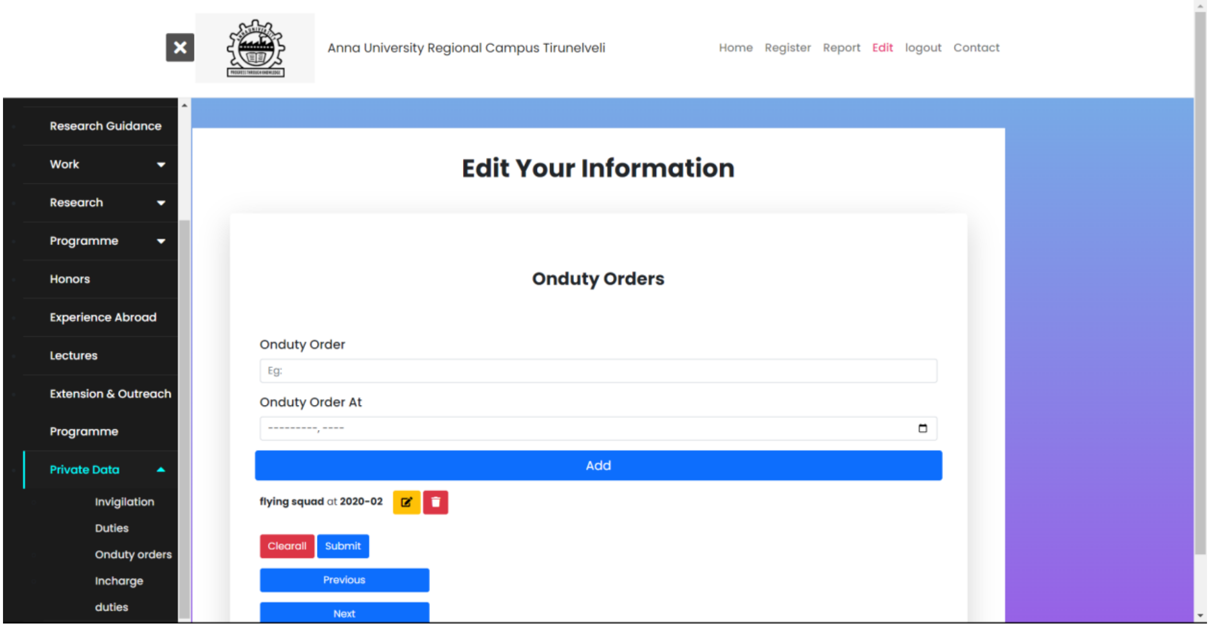


Fig 5.1.17 Faculty Profile Maintenance private data

The edit page information’s can be classified as two types such as the information contains public data and the private data

The fig 5.1.17 show the information that contains private data the private data cannot be viewed by anyone and has some restriction. The private data can be accessed by only the admin and the faculty members the private data cannot be accessed by any other persons.

Faculty Profile Maintenance Private data:



Edit Your Information

Onduty Orders

Onduty Order: Eg:

Onduty Order At: ---------, ----

flying squad at 2020-02

Fig 5.1.18 Faculty Profile Maintenance private data

The onduty order form page of the faculty profile maintenance will contain the private data.

The fig 5.1.18 show the information that contains private data of onduty duty for the faculty member who works in the institution, the private data cannot be viewed by anyone and can only be viewed by the admin and the respective faculty member.

5.2 TEST CASES:

Test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement. Test cases underlie testing that is methodical rather than haphazard. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing.

5.2.1 TEST CASE FOR PROFILE VIEW

Test Case ID	**Test Objective**	**Pre-Conditions**	**Test Steps**	**Expected output Positive Testcase**	**Expected output Negative Testcase**	**Actual Output**	**Status Pass/Fail**
TC_01	To select the department	User should be in the home page and select the department from the dropdown bar	Click on department and select the department	Faculty members names should be listed down	Faculty members names will not be listed down	Faculty members names listed down	pass
TC_02	To test the registration feature	User should be on the register page	Enter the details to register as a new user	The user should be registered as a faculty member	The user should not be registered as a faculty member	User register as a faculty member	pass
TC_03	To verify email verification	User should log into their mail account to verify the email	User should verify that the mail belongs to him/her	Email should be verified	Email should not be verified	Email is verified	pass
TC_04	To test the login feature	User should be on the login page	Enter the login credentials (staff ID and password)	Staff ID and password should be accepted	Staff ID and password should not be accepted	Staff ID and password is accepted	pass

TC_05	To test the functionality of adding details to the form	User must be logged in as a faculty member	Add information to their form And press submit	Data provided will be accepted and stored	Data provided will not be accepted and stored	Data provided will be accepted and stored	pass
TC_06	To test the functionality of editing details to the form	User must be logged in as a faculty member	Edit information in the existing form and press submit	Data provided will be accepted and stored	Data provided will not be accepted and stored	Data provided will be accepted and stored	pass
TC_07	To test the functionality of de details to the form	User must be logged in as a faculty member	Delete information in the existing form and press submit	Data provided will be accepted and stored	Data provided will not be accepted and stored	Data provided will be accepted and stored	pass
TC_07	To test the feature of viewing faculty details	User should be in the home page and select the department from the dropdown bar	Click on view on the specific form to view the details of the faulty	Details of the particular form chosen by user should be displayed	Details of the particular form chosen by user should not be displayed	Details of the particular form chosen by user is displayed	pass
TC_08	To test the aspects of printing the form	User should in the specific form	Click on the Print to PDF button on a particular form	PDF of the form will be downloaded	PDF of the form will not be downloaded	PDF of the form is downloaded	pass
TC_09	To test the functionality of generating report	User must be logged in as Admin	Enter the time period and the department And press submit	Data will be provided	Data will not be provided	Data will be provided	pass

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENT

6.1 CONCLUSION

Thus, the project Faculty Profile Maintenance is completed successfully with functional components and requirements satisfied. This application helps the staffs to view their information in the form of pdf and can download the pdf. This web-based application is designed using HTML and Bootstrap and is powered by backend technologies like PHP and MySQL with the help of XAMPP server.

6.2 FUTURE ENHANCEMENT

The future enhancements that can be implemented in this project is to make this an independent module that can run on a production environment with a master database connected to it.

APPENDIX

SAMPLE CODING

Code for Educational Qualification Form

```
const edq_form = document.getElementById("edq_form");
const degree = document.getElementById("degree");
const degree_from = document.getElementById("degree_from");
const degree_to = document.getElementById("degree_to");
const edqs = document.getElementById("edqs");
const add_edq = document.getElementById("add_edq");
const alert = document.querySelector(".alert");
const clear_all = document.getElementById("clear_all");
const previous = document.getElementById("previous");
const next = document.getElementById("next");

let edit_flag = false;
let edit_element;
let edit_tag;
let edit_from;
let edit_to;
let edit_id = "";

// -------------------------------------------- add / edit edq  -------------------------------
------------- //
// Generate unique ID
function uuid() {
   return "xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx".replace(/[xy]/g,
function(c) {
       var r = (Math.random() * 16) | 0,
           v = c == "x" ? r : (r & 0x3) | 0x8;
       return v.toString(16);
   });
}

// !!!Alert!!!
function display_alert(text, action) {
   alert.textContent = text;
   alert.classList.add(`alert-${action}`);
   // remove alert
   setTimeout(function() {
       alert.textContent = "";
       alert.classList.remove(`alert-${action}`);
   }, 4000);
}

// set backt to defaults
function set_back_to_default() {
   degree.value = "";
   degree_from.value = "";
   degree_to.value = "";
   edit_flag = false;
   edit_id = "";
   add_edq.textContent = "Add";
}

// delete an item
function delete_item(e) {
   const element = e.currentTarget.parentElement.parentElement;
   const id = element.dataset.id;

   edqs.removeChild(element);
   display_alert("item removed", "danger");

   set_back_to_default();
   // remove from local storage
   remove_from_local_storage(id);
}

// edit an item
async function edit_item(e) {
   const element = e.currentTarget.parentElement.parentElement;
   // set edit item
   edit_tag = element;
   edit_element = e.currentTarget.previousElementSibling;
   // set form value
   degree.value = edit_element.childNodes[0].innerHTML;
   degree_from.value = edit_element.childNodes[2].innerHTML;
   degree_to.value = edit_element.childNodes[4].innerHTML;
   // faculty.value = edit_fac;
   edit_flag = true;
   edit_id = element.dataset.id;
   //
   add_edq.textContent = "Edit";
}

// clear items
```

```
function clear_items(e) {
    e.preventDefault();

    return new Promise((resolve, reject) => {
        // confirm clear
        if (!window.confirm("Are you sure to clear all?")) {
            return;
        }

        window.localStorage.removeItem("edq");
        const items = document.querySelectorAll(".one-edq");
        if (items.length > 0) {
            items.forEach(function(item) {
                edqs.removeChild(item);
            });
        }
        display_alert("removed all educational qualifications", "danger");
        resolve(set_back_to_default());
    });
}

// add item to the list
function add_edqs(e) {
    e.preventDefault();

    // **************************************** create
**************************************** //
    if (degree.value && degree_from.value && degree_to.value && !edit_flag)
{
        const id = uuid();
        const element = document.createElement("article");
        let attr = document.createAttribute("data-id");
        attr.value = id;

        element.setAttributeNode(attr);
        element.classList.add("one-edq");

        element.innerHTML = `
      <p class="one-degree"><span><b>${degree.value}</b> from
<b>${degree_from.value}</b> to <b>${degree_to.value}</b></span>
      &ensp;
            <button type="button" class="edit-btn btn btn-warning">
              <i class="fas fa-edit"></i>
            </button>
            <button type="button" class="delete-btn btn btn-danger">
              <i class="fas fa-trash"></i>
            </button></p>
          `;
        // add event listeners to both buttons;
        const deleteBtn = element.querySelector(".delete-btn");
        deleteBtn.addEventListener("click", delete_item);
        const editBtn = element.querySelector(".edit-btn");
        editBtn.addEventListener("click", edit_item);

        // append child
        edqs.appendChild(element);
        // display alert
        display_alert("Added Successfully", "success");
        // set local storage
        add_to_local_storage(id, degree.value, degree_from.value,
degree_to.value);
        // // set back to default
        set_back_to_default();
        // ****************************************** Edit
****************************************** //
    } else if (
        degree.value &&
        degree_from.value &&
        degree_to.value &&
        edit_flag
    ) {
        edit_element.innerHTML = `<b>${degree.value}</b> from
<b>${degree_from.value}</b> to <b>${degree_to.value}</b>`;
        display_alert("values changed", "success");

        // edit  local storage
        edit_local_storage(
            edit_id,
            degree.value,
            degree_from.value,
            degree_to.value
        );
        set_back_to_default();
    } else {
        display_alert("please fill all the fields", "danger");
    }
}
```

```
// -------------------------------------------- Local Storage -------------------------------------------- //
// get eqd
const get_user = () => {
    return window.localStorage.getItem("user") ?
        JSON.parse(window.localStorage.getItem("user")) :
        [];
};

// get eqd
const get_local_storage = () => {
    return window.localStorage.getItem("edq") ?
        JSON.parse(window.localStorage.getItem("edq")) :
        [];
};

// add item to local storage
const add_to_local_storage = (id, deg, deg_f, deg_t) => {
    const item = {
        degree_id: id,
        degree: deg,
        degree_from: deg_f,
        degree_to: deg_t,
    };
    let items = get_local_storage();
    items.push(item);
    window.localStorage.setItem("edq", JSON.stringify(items));
};
```

```
// remove from local storage
function remove_from_local_storage(id) {
    let items = get_local_storage();

    items = items.filter((item) => {
        if (item.degree_id != id) {
            return item;
        }
    });

    window.localStorage.setItem("edq", JSON.stringify(items));
}

// edit an element in local storage
function edit_local_storage(id, deg, deg_f, deg_t) {
    let items = get_local_storage();

    items = items.map(function(item) {
        if (item.degree_id == id) {
            item.degree = deg;
            item.degree_from = deg_f;
            item.degree_to = deg_t;
        }
        return item;
    });
    window.localStorage.setItem("edq", JSON.stringify(items));
    return;
}
```

```
// -------------------------------------- Setup Items after refresh -----------------------
--------------- //

// get from local storage
function setup_items() {
    edqs.innerHTML = "";
    let items = get_local_storage();
    // console.log("local storage", items);
    if (items.length > 0) {
        items.forEach(function(item) {
            create_list_item(
                item.degree_id,
                item.degree,
                item.degree_from,
                item.degree_to
            );
        });
    }

    set_back_to_default();
}

// append te child element to html
async function create_list_item(id, deg, deg_f, deg_t) {
    const element = document.createElement("article");
    let attr = document.createAttribute("data-id");
    attr.value = id;
```

```
  element.setAttributeNode(attr);
  element.classList.add("one-edq");

  element.innerHTML = `
   <p class="one-degree"><span><b>${deg}</b> from <b>${deg_f}</b> to
<b>${deg_t}</b></span>
   &ensp;
        <button type="button" class="edit-btn btn btn-warning">
         <i class="fas fa-edit"></i>
        </button>
        <button type="button" class="delete-btn btn btn-danger">
         <i class="fas fa-trash"></i>
        </button></p>
      `;
  // add event listeners to both buttons;
  const deleteBtn = element.querySelector(".delete-btn");
  deleteBtn.addEventListener("click", delete_item);
  const editBtn = element.querySelector(".edit-btn");
  editBtn.addEventListener("click", edit_item);

  // append child
  edqs.appendChild(element);
}

// ------------------------------------------ Submit the form  ---------------------------
---------------- //
function submit_form(e) {
  e.preventDefault();
```

```
let educational_qualification = [];

const edq = get_local_storage();

// // check if empty
// if (edq.length <= 0) {
//   display_alert("please add some educational qualifications", "danger");
//   return;
// }

edq.forEach((item) => {
   item.degree_id = isNaN(Number(item.degree_id)) ? 0 : item.degree_id;
   educational_qualification.push(item);
});
const user = get_user();

if (!user) {
   window.alert("user not logged in");
   return;
}
const xhr = new XMLHttpRequest();

xhr.open(
   "POST",
   `../../api/profile/public/type_5/degree.php?ID=${user.user_id}`,
   true
);

xhr.onreadystatechange = function() {
        if (xhr.readyState == XMLHttpRequest.DONE) {
            const got = JSON.parse(xhr.responseText);

            if (got.error) {
                display_alert(got.error, "danger");
            } else {
                window.alert("Educational Qualifications updated successfully");
                got.forEach((item, index, array) => {
                    item.degree_from = item.degree_from.substr(0, 7);
                    item.degree_to = item.degree_to.substr(0, 7);

                    if (index + 1 == array.length) {
                        // assign the data
                        window.localStorage.setItem("edq", JSON.stringify(got));
                        resolve(setup_items());
                    }
                });
            }
        }
    };

    xhr.send(JSON.stringify(educational_qualification));
}

// -------------------------------------------- Initially  ------------------------------------
-------- //
// DB data
const db_data = () => {
    return new Promise(async(resolve, reject) => {
```

```
const user = get_user();
if (!user) {
    window.alert("user not logged in");
    return;
}

const xhr = new XMLHttpRequest();

xhr.open(
    "GET",
    `../../api/profile/public/type_5/degree.php?ID=${user.user_id}`,
    true
);

xhr.onreadystatechange = function() {
    if (xhr.readyState == XMLHttpRequest.DONE) {
        const got = JSON.parse(xhr.responseText);

        if (got.error) {
            // if can't get the data, thorw the error
            reject(display_alert(got.error, "danger"));
        } else {
            // change date
            got.forEach((item, index, array) => {
                item.degree_from = item.degree_from.substr(0, 7);
                item.degree_to = item.degree_to.substr(0, 7);

                if (index + 1 == array.length) {
                    // assign the data
```

```
                            window.localStorage.setItem("edq", JSON.stringify(got));
                            resolve(setup_items());
                        }
                    });
                }
            }
        };
        xhr.send();
    });
};
// Initial setup
async function initialize() {
    if (get_local_storage().length !== 0) {
        setup_items();
    } else {
        db_data();
    }
}

// add an edq
add_edq.addEventListener("click", add_edqs);
// when form submitted
edq_form.addEventListener("submit", submit_form);
// initialize
window.addEventListener("DOMContentLoaded", initialize);
// clear all
clear_all.addEventListener("click", clear_items);
// previous button
previous.addEventListener("click", () => {
    window.localStorage.removeItem("edq");
    window.location.replace("./edit2.html");
});
// next button
next.addEventListener("click", () => {
    window.localStorage.removeItem("edq");
    window.location.replace("./position.html");
});
```

REFERENCES

1. **https://www.w3schools.com**
2. **https://www.youtube.com/**
3. **https://developer.mozilla.org/en-US/docs/Web/HTML**
4. **https://developer.mozilla.org/en-US/docs/Web/CSS]**
5. **https://developer.mozilla.org/en-US/docs/Web/javascript]**
6. **https://developer.mozilla.org/en-US/docs/Glossary/Bootstrap**
7. **https://developer.mozilla.org/en-US/docs/Glossary/PHP**
8. **https://developer.mozilla.org/en-US/docs/Glossary/SQL**
9. **https://en.wikipedia.org/wiki/MySQL**