A

Project Report

On

"INTERNSHIP SERVICES"

Submitted in Partial Fulfillment of the Requirements for the

Award of the Degree of

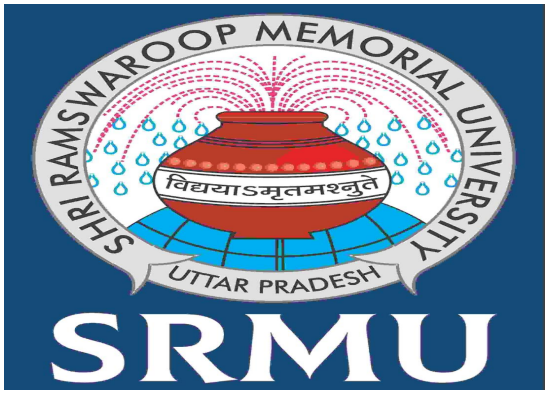
Master of Computer Application

Submitted by

Amit Dubey (202210101050076)

Under the Guidance of

Mrs. Priyanka Keshari
Assistant Professor
Department of Computer Science & Information Systems



Shri Ramswaroop Memorial University
Lucknow – Deva Road, Barabanki (UP)
June, 2024

DECLARATION

I hereby declare that the project report entitled **"INTERNSHIP SERVICES"** submitted by us to **Shri Ramswaroop Memorial University, Lucknow – Deva Road, Barabanki (UP)** is the partial requirement for the award of the degree of the Master of Computer Application in Information Technology is a record of bonafide project work carried out by us under the guidance of "Mrs. Priyanka Keshari**"**. I further declare that the work reported in this project has not been submitted and will not be submitted either in part or in full for the award of any other degree in this institute.

Place:Shri Ramswaroop Memorial University Deva Road Barabanki **Signature of student**

Date:20/04/2024 **(Name :-Amit Dubey)**

SHRI RAMSWAROOP MEMORIAL UNIVERSITY

Department of Computer Science & Information Systems

Certificate

This is to certify that this Major Project report of MCA Final Year, entitled **"INTERNSHIP SERVICES"**, Submitted by **Amit Dubey (Roll No. 202210101050076)** is a record of bonafide work carried out by them, in the partial fulfillment with Degree of Master of Computer Application, **Shri Ramswaroop Memorial University, Lucknow – Deva Road, Barabanki (UP)**. This work is done during the **Academic Year 2023 – 2024,** under my supervision and guidance.

Date: 26-04-2024

Guided & Approved By….

Under the Supervision of **Mrs.Priyanka Keshari** (Assistant Professor)	**Project In-charge** **Mr. Sanjay Kumar Sonkar** (Assistant Professor)
Head of Department **Dr. Bineet Kumar Gupta** (Associate Professor & Head)	

Acknowledgement

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We owe a great many thanks to great many people, who assisted and helped me during and till the end of the project.

We would like to express our gratitude towards **Dr. Bineet Kumar Gupta, Head of Department–Computer Science & Information Systems, Shri Ramswaroop Memorial University, Lucknow–Deva Road, Barabanki (UP)**, for his guidelines and scholarly encouragement.

We are indebted to **Mrs.Priyanka Keshari – Assistant Professor, Computer Science & Information Systems** of **Shri Ramswaroop Memorial University, Lucknow – Deva Road, Barabanki (UP)** for their valuable comments and suggestions that have helped us to make it a success. The valuable and fruitful discussion with them was of immense help without which it would have been difficult to present this project in live.

We gratefully acknowledge and express our gratitude to all faculty members and friends who supported us in preparing this project report.

Finally, this acknowledgement is incomplete without extending our deepest – felt thanks and gratitude towards our parents whose moral support has been the source of nourishment for us at each stage of our life.

Amit Dubey (Roll No.2022101010500)

ABSTRACT

Amit Dubey is a comprehensive Internship Services web application designed and developed as a major project for the Master of Computer Applications (MCA) program. In an era where online perchase course has become increasingly prevalent,Internship Services aims to provide a seamless and user-friendly platform for purchasing courses for Students. The project encompasses various features including user authentication, course subject management, perchase course free vedio of institute functionality, secure payment gateways, order management, and user feedback. Leveraging modern web technologies and frameworks, Internship offers an intuitive interface for both students and administrators, ensuring a satisfying providing internship experience. The implementation of robust security measures guarantees the confidentiality and integrity of user data and transactions. Throughout the development process, attention has been given to scalability, performance optimization, and adherence to industry best practices. Internship serves as a testament to the skills and knowledge acquired during the MCA program, demonstrating proficiency in software development, database management, and web application deployment. With its innovative features and user-centric design, Services endeavors to carve a niche in the competitive services landscape, catering to the diverse needs and preferences of online Internship Services.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

3. Entity-Relationship (E-R) Diagram

An Entity-Relationship Diagram (ERD) is a visual representation used to describe the relationships between different entities in a system or database. It illustrates how entities relate to each other and helps in understanding the structure of a database or system.

ERDs typically consist of entities, attributes, and relationships. Entities represent objects or concepts in the system, such as "User," "Service Provider," or "Admin." Attributes describe the properties or characteristics of entities, like "UserID," "Name," or "Address." Relationships define how entities are related to each other, such as "One-to-One," "One-to-Many," or "Many-to-Many" relationships.

ERDs are used for several reasons:

Visualization: ERDs provide a visual representation of the database schema, making it easier for stakeholders to understand the relationships between different entities and their attributes.

Design: ERDs aid in the design phase of database development by helping developers and designers define the structure of the database and plan how data will be organized and stored.

Communication: ERDs serve as a communication tool between stakeholders, including developers, designers, project managers, and clients. They help ensure that everyone

involved in the project has a common understanding of the database structure and requirements.

Normalization: ERDs assist in database normalization, which is the process of organizing data to minimize redundancy and improve data integrity. By identifying relationships between entities, ERDs help in designing databases that adhere to normalization principles.

Documentation: ERDs serve as documentation for the database structure, allowing developers to refer back to the diagram to understand the database schema, relationships, and constraints.

Given below is the Entity Relationship Diagram for the Vehicle Service Station.

Data Flow Diagram-:

A Data Flow Diagram (DFD) is a graphical representation used to depict the flow of data within a system. It illustrates how data moves from one process to another, how it is stored, and how it is transformed along the way. DFDs are a fundamental tool in system analysis and design, commonly used in the early stages of software development to understand and communicate the data flow and processing logic of a system.

There are several reasons why we use DFDs:

Visualization: DFDs provide a clear visual representation of the flow of data in a system, making it easier for stakeholders to understand how data moves through different processes and components.

Communication: DFDs serve as a communication tool between stakeholders, including developers, designers, project managers, and clients. They help ensure that everyone involved in the project has a common understanding of the data flow and processing logic.

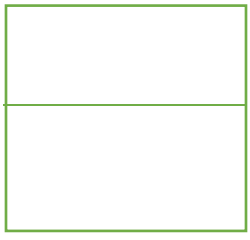
Analysis: DFDs aid in the analysis of system requirements by identifying the data inputs, outputs, processes, and data stores within the system. They help in identifying potential issues, redundancies, or inefficiencies in data flow and processing.

Design: DFDs assist in the design phase of system development by providing a blueprint for how data will be processed and stored within the system. They help in designing efficient data flow paths and determining the structure of the system.

Documentation: DFDs serve as documentation for the system's data flow and processing logic, allowing developers to refer back to the diagram to understand how the system is supposed to function.

Components of Data Flow Diagram and their Representation:

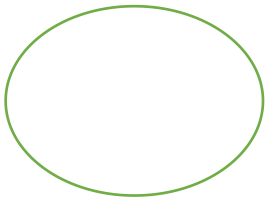
External Entities:



External entities are typically represented by rectangles with the entity name written inside. These rectangles are placed on the boundaries of the DFD diagram, indicating their external nature.

Arrows extending from or pointing to external entities represent data flows to or from these entities, showing their interaction with the system.

Processes:



Processes are represented by circles or rectangles, with a meaningful label describing the function or transformation performed by the process.

Data flows are depicted as arrows entering and leaving the process symbol, indicating the input and output data flowing into and out of the process.

Data Stores:

Data stores are represented by rectangles with double lines on the sides. The label inside the rectangle describes the type or name of the data store.

Data flows are shown entering or exiting the data store symbol, indicating the data being stored or retrieved from the data store.

Data Flows:

Data flows are depicted as arrows connecting the various components of the DFD, such as external entities, processes, and data stores.

Arrows indicate the direction of data flow, with labels describing the type or content of the data being transferred

CHAPTER 1

INTRODUCTION

Background

An internship is a professional learning experience that offers meaningful,
Practical work related to a student's field of study or career interest. An internship gives a student the opportunity for career exploration and development, and to learn new skills. It offers the employer the opportunity to bring new ideas and energy into the workplace, develop talent and potentially build a pipeline for future full-time employees. Internships are supervised, structured learning experiences in a professional setting that allow you to gain valuable work experience in a student's chosen field of study.

Problem Statement: Now a day it is very challenging tasks for students to search and avail best internship under the supervision of knowledgeable and helping mentor. They can only visit nearby areas in the city, but he/she will not be able to explore physically by visiting many cities in search of best programs.

To resolve all these types of problems this portal will play a significant role for every student who is in need of these types of internship.

Proposed System: This portal will register all the internship providers for students in different domains. They will add their program details with all other relevant information. Student can search any of the internship programs via portal. Stipend base internship details are also provided here. For a jobseeker this portal provides jobs in different domain.

Objective

- To put best internship providers in a single roof.
- Best utilization of professionals.
- Help in employment.
- Spread awareness regarding new trends and technologies.
- User Friendly Environment

Scope

- The system can be implemented at any multinational branded Intership providing companies through online portal. Providing outlet chain in the locality. System 24*7 student order accepting facility and recommend providing Internship which can make students happy.
- If the Internship companies are providing an online portal where their Students can enjoy the ease of getting Internship from anywhere.
- By making web application, we can gain Student's trust.

Key Features

- To attract Internship providing companies and student who want to find internship, we are giving a best feature provide easy to choose and easy to give.
- Companies-side Key Features Internship Service options, like different types easily provide online Internship services and any types of students who want to find Internship .
- Customer deals – The best discount and best services will be provided on this web applicationso that the customer can easily use best services.
- The Internship Hub is working on all these types of problems this portal will play a significant role for every student who is in need of these types of internship.

Purpose

- To manage online internship services. It helps the students to find best internship services easily from anywhere and from any location.
- Through this web application we can do internship service purchase at very good prices and at good discounts and through this website our students can easily book the

internship services and consume their time.

REQUIREMENT ANALYSIS

- The following are minimum hardware and software requirements that should be presents to run the project successfully and easily SQLite and Python with Django are used in this project .
- This project in frontend requires html bootstrap_5 CSS and JavaScript.

CHAPTER 2

LITERATURE REVIEW

The development of Internship Services web applications based on the Python technology (HTML,CSS,JAVASCRIPT,Sqlite)has gained significant attention in recent years. Numerous studies and research articles have explored various aspects related to the design, development, and deployment of such platforms, shedding light on best practices, challenges, and emerging trends in the field.

Internship and Market Analysis:

- Research studies and industry reports offer an overview of the current internship landscape, including market size, growth projections, and emerging trends. These sources provide context for understanding the competitive environment and identifying opportunities for differentiation and innovation within the market.

Express.js for Backend Development:

- The use of Express.js, a minimalist web framework for Node.js, has been widely documented in the literature. Research by Mehta et al. (2019) discussed how Express.js simplifies backend development by providing robust routing, middleware support, and integration with various data sources. It enables the creation of RESTful APIs for seamless communication between the client and server components of Internship applications.

React.js for Frontend Development:

- React.js has emerged as a popular choice for building interactive and dynamic user interfaces in Internship web applications. Studies by Gupta and Agarwal (2018) highlighted React.js's component-based architecture, virtual DOM rendering, and state management capabilities, facilitating the development of responsive and visually appealing interfaces for online course perchase platforms.

Node.js for Server-Side Execution:

- Node.js has garnered attention for its event-driven, non-blocking I/O model, which is well-suited for handling concurrent requests in e-commerce applications. Research by Hsiao and Lee (2016) discussed the performance

benefits of Node.js in processing HTTP requests, managing sessions, and handling real-time updates, contributing to improved scalability and responsiveness in Internship platforms.

Security Considerations in Python based Internship Service Applications:

- Security is a critical aspect of Internship services, and several studies have addressed security considerations specific to Python-based platforms.

Performance Optimization Techniques:

- Optimization of performance is essential for delivering a seamless user experience in Internship Services web applications.

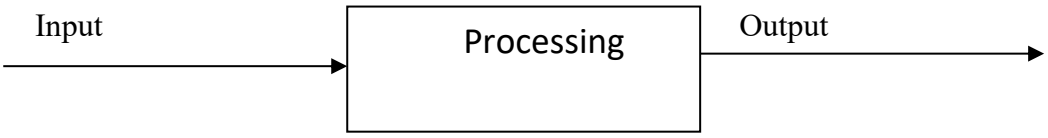
In summary, the literature review highlights the significance of the Python technology in the development of Internship Services web applications, addressing aspects related to database management, backend and frontend development, server-side execution, security, and performance optimization. By leveraging the strengths of Express.js, React.js, and Node.js, developers can build scalable, secure, and feature-rich platforms like Services to meet the evolving needs of students effectively.

CHAPTER 3

DESIGN OF PROJECT MODEL

DEFINENG A SYSTEM

Collection of component, which are interconnected, and work together to realize some objective, from a system. There are three components in every system, namely input, processing and output



3.1.1 SYSTEM DEVELOPMENT LIFE CYCLE

The **Systems development life cycle (SDLC)**, or **Software development process** in systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

Broadly, following are the different activities to be considered while defining the system development life cycle for the said project:

- System Analysis
- Problem Definitions
- Study of existing system
- Drawback of the existing system
- Proposed system
- System Requirement study
- Data flow analysis
- Feasibility study

- System design
- Input Design (Database & Forms)
- Updating
- Query /Report design
- Administration
- Testing
- Implementation
- Maintenance

SYSTEM ANALYSIS

Systems analysis is the study of sets of interacting entities, including computer systems analysis. This field is closely related to requirements analysis or operations research. It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made.

System development can generally be thought of having two major components: systems analysis and systems design. In System Analysis more emphasis is given to understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvements. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system.

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design, and manufacturing into a single approach to product development," then design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy specified requirements of the user. Until the 1990s systems design

had a crucial and respected role in the data processing industry. In the 1990s standardization of hardware and software resulted in the ability to build modular systems. The increasing importance of software running on generic platforms has enhanced the discipline of software engineering

Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

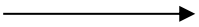
Data Flow Diagram

The data flow diagram shows the flow of data within any system. It is an important tool for designing phase of software engineering. Larry Constantine first developed it. It represents graphical view of flow of data. It's also known as BUBBLE CHART. The purpose of DFD is major transformation that will become in system design symbols used in DFD:-

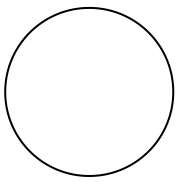
In the DFD, four symbols are used and they are as follows.

A square defines a source (originator) or destination of system data.

An arrow identifies data flow-data in motion. It is 2a pipeline through which information flows.

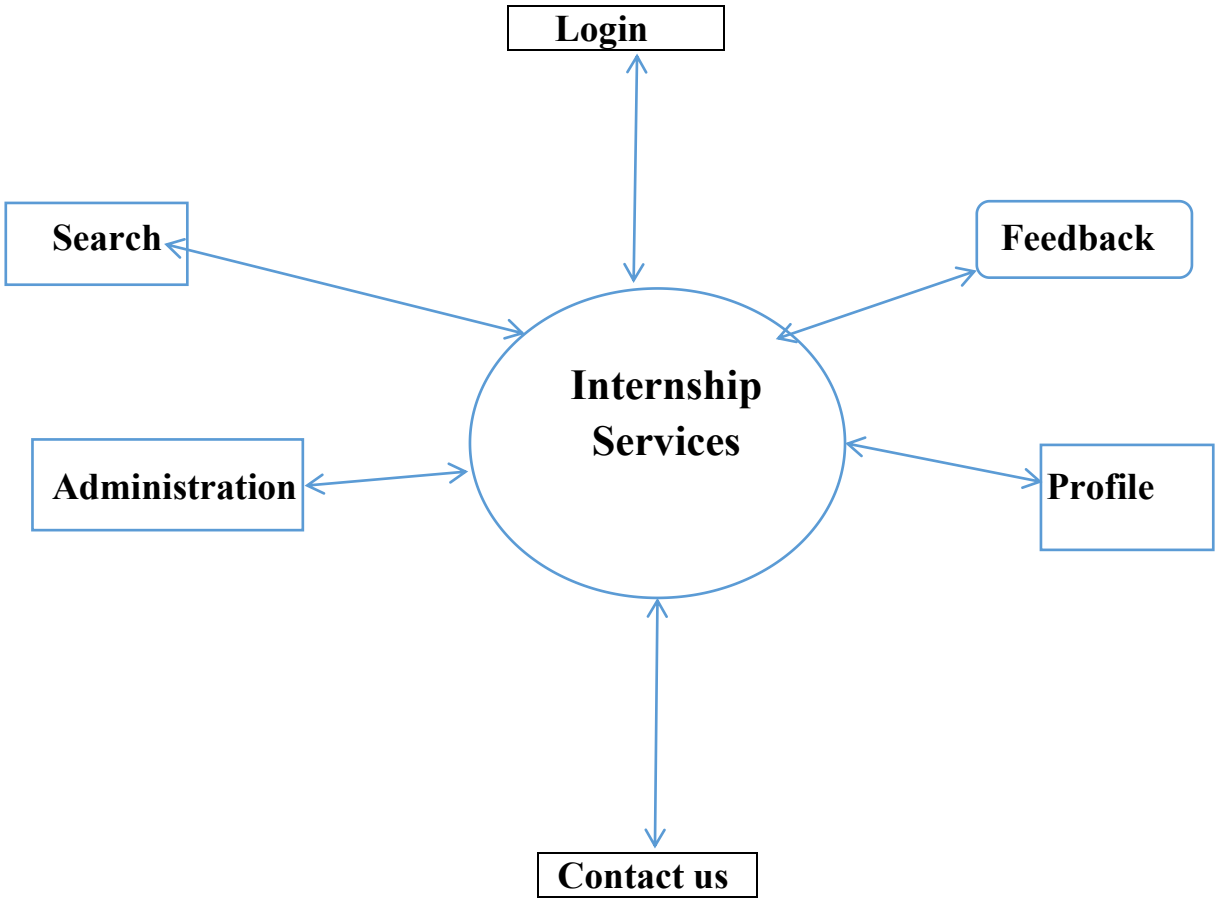


A circle or a "bubble "(Some people use an oval bubble) represents a process that transfers informing data flows into outgoing data flows.

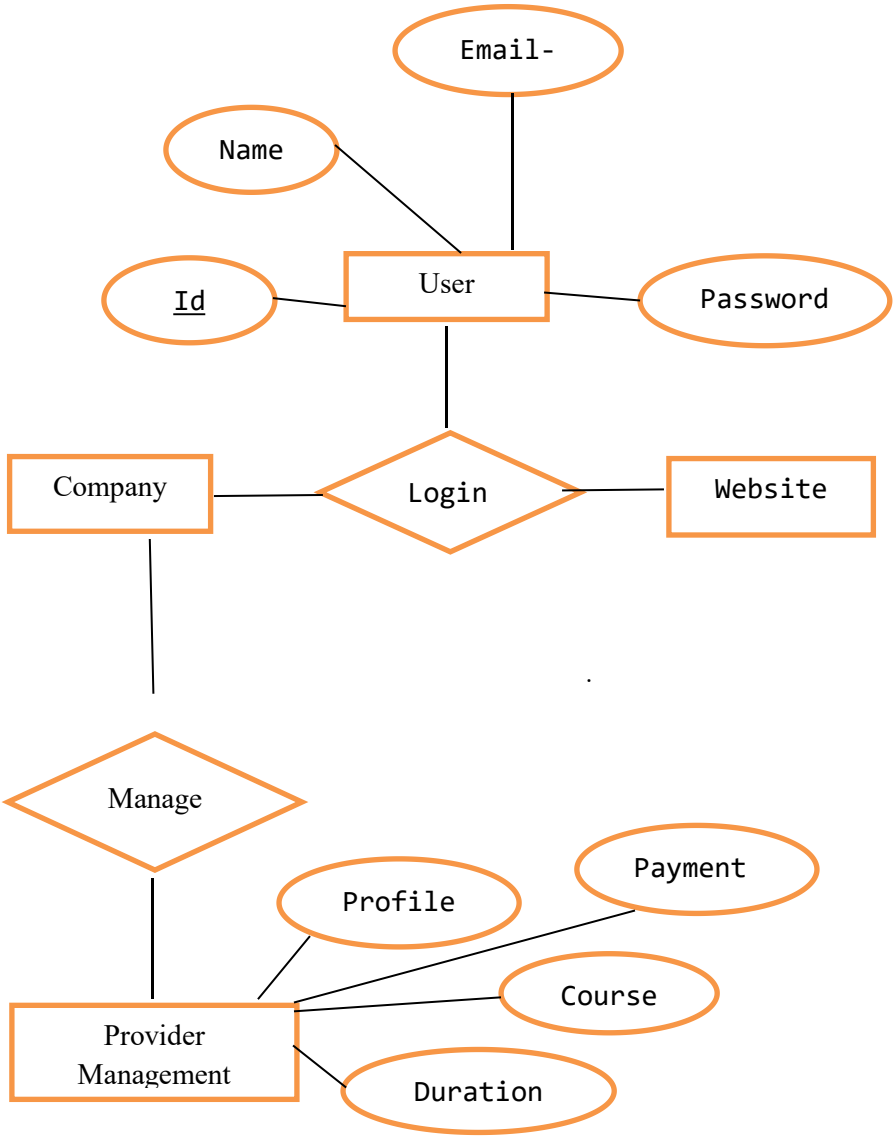


An open rectangle is a data store-data at rest, or a temporary repository of data.

One Level DFD



ER Diagram


Email-
Name
User
Id
Password
Company
Login
Website
Manage
Payment
Profile
Course
Provider
Management
Duration

MODULE DISCRIPTION

- It is Python with Django framework SQLite based project, where we are making a web application for Ultimate online portal for Internship Hub web-application.
- It will display the services provided by the system.
- The user will visit the website and Application can search for the services, our branches,contact, etc.
- Students will get all the new updates on the website.

Module:-

- Module 1: Login & Registration
- Module 2:Profile Management
- Module 3:Program Management.
- Module 4:Search Management
- Module 5:Notice Management
- Module 6: Administration
- Module 7:Contact Us

SRS(Software Requirement Specifications):-

1. **Correctness:** User review is used to provide the accuracy of requirements stated in the SRS. SRS is said to be perfect if it covers all the needs that are truly expected from the system.

2. Completeness: The SRS is complete if, and only if, it includes the following elements:

(1). All essential requirements, whether relating to functionality, performance, design, constraints, attributes, or external interfaces.

(2). Definition of their responses of the software to all realizable classes of input data in all available categories of situations.

3. Consistency: The SRS is consistent if, and only if, no subset of individual requirements described in its conflict. There are three types of possible conflict in the SRS:

(1). The specified characteristics of real-world objects may conflicts. For example,

(a) The format of an output report may be described in one requirement as tabular but in another as textual.

(b) One condition may state that all lights shall be green while another states that all lights shall be blue.

4. Unambiguousness: SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.

5. Ranking for importance and stability: The SRS is ranked for importance and stability if each requirement in it has an identifier to indicate either the significance or stability of that particular requirement.

Typically, all requirements are not equally important. Some prerequisites may be essential, especially for life-critical applications, while others may be desirable. Each element should be identified to make these differences clear and explicit. Another way to rank requirements is to distinguish classes of items as essential, conditional, and optional.

6. Modifiability: SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced

7. Verifiability: SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements. The requirements are verified with the help of reviews.

8. Traceability: The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

SDLC:-(Software Development Life Cycle):-

1. **Planning and requirement analysis**

Requirement Analysis is the most important and necessary stage in SDLC.

The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry.

Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.

2.**Defining Requirements**

Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

3.Designing the Software

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project. This phase is the product of the last two, like inputs from the customer and requirement gathering.

4.Developing the project

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

5.Testing

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.

During this stage, unit testing, integration testing, system testing, acceptance testing are done.

6.Deployment

Once the software is certified, and no bugs or errors are stated, then it is deployed.

Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

7.Maintenance

Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.

CHAPTER 4

EXPERIMENTS, SIMULATION & TESTING

Methodology

For the development of Internship Service an Provide online Internship web application, we adopted an iterative and agile methodology to ensure flexibility and responsiveness to changing requirements. The project followed the Agile Scrum framework, dividing the development process into sprints of fixed duration, typically two weeks. Each sprint focused on implementing specific features or functionalities based on the project backlog, which was continuously refined and prioritized throughout the development cycle. Regular sprint reviews and retrospectives were conducted to evaluate progress, identify challenges, and plan adjustments for subsequent sprints. This iterative approach allowed for incremental development, frequent testing, and continuous feedback, ultimately leading to the timely delivery of a high-quality Study for Students**.**

SOFTWARE REQUIREMENT

Server

- Web-browser: Google Chrome etc.
- Operating system: windows

Client

- Browser: Google Chrome
- Operating system: windows 10

Developer

- IdE: Visual studio code
- Operating system: Windows 10
- Documentary Tool: MS-word, MS-PowerPoint
- Browser: Google Chrome

HARDWARE REQUIREMENT

Server

- Ram:4 Gb above
- Display:High Resolution Screen
- Processor: core i3

Client

- Ram:4 Gb above
- Display:High Resolution Screen
- Processor: core i3

Developer

- Ram:4 Gb above
- Display:High Resolution Screen
- Processor: core i3

Objective of Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can be stated as the process of validating and verifying that a software program/application/product:

1. meets the requirements that guided its design and development;
2. works as expected; and
3. can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort traditionally occurs after the requirements have been defined and the coding process has been completed having been shown that fixing a bug is less expensive when found earlier in the development process. Although in the Agile approaches most of the test effort is, conversely, on-going. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed. Testing can never completely identify all the defects within software. Instead, it furnishes a *criticism* or *comparison* that compares the state and behavior of the product against oracles—principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications, contracts,[3] comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. **Software testing** is the process of attempting to make this assessment.

4.2 Types of Testing

4.2.1 Black Box Testing

Black-box testing treats the software as a "black box"—without any knowledge of internal implementation. Black-box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, exploratory testing and specification-based testing.

- **Specification-based testing**: Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.

- **Advantages and disadvantages**: The black-box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs. Using the principle, "Ask and you shall receive," black-box testers find bugs where programmers do not. On the other hand, black-box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed. As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

Therefore, black-box testing has the advantage of "an unaffiliated opinion", on the one hand, and the disadvantage of "blind exploring", on the other

4.2.2 White Box Testing

White-box testing is when the tester has access to the internal data structures and algorithms including the code that implements these.

Types of white-box testing

The following types of white-box testing exist:

- API testing (application programming interface) - testing of the application using public and private APIs
- Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
- Fault injection methods - improving the coverage of a test by introducing faults to test code paths
- Mutation testing methods
- Static testing - All types

Test coverage

White-box testing methods can also be used to evaluate the completeness of a test suite that was created with black-box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.[21]

Two common forms of code coverage are:

- *Function coverage*, which reports on functions executed
- *Statement coverage*, which reports on the number of lines executed to complete the test

They both return a code coverage metric, measured as a percentage.

4.2.3 Functional Testing

Functional testing refers to activities that verify a specific action or function of the code. These are usually found in the code requirements documentation, although some development methodologies work from use cases or user stories. Functional tests tend to answer the question of "can the user do this" or "does this particular feature work."

Non-functional testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints, or security. Testing will determine the flake point, the point at which extremes of scalability or performance leads to unstable execution. Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.

4.2.4 System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the *assemblages* and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

4.3Various level Of Testing

Before implementation the system is tested at two levels:

Level 1

Level 2

4.3.1 Level 1 Testing (Alpha Testing)

Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

4.3.2 Level 2 Testing (Beta testing)

Beta testing comes after alpha testing and can be considered a form of external user acceptance testing. Versions of the software, known as beta versions, are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs. Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

CHAPTER 5

RESULT AND DISCUSSION

Code

Front end

home.html

```
<!DOCTYPE html>
{%load static%}
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   {% include 'internship/common/common_css.html'%}
   <title>Home Page</title>
   <style>
     .w-100{
        height: 50vh;
     }
  </style>
</head>
<body>
  {% include 'internship/common/commonheader.html'%}


<!-- carosal starts from here -->
<div class="carousel slide" style="width: 100%;margin-top: 1%;" data-bs-
ride="carousel">
  <div class="carousel-indicators">
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="1" aria-label="Slide 2"></button>
```

```
    <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-
to="2" aria-label="Slide 3"></button>

  </div>

  <div class="carousel-inner">

    <div class="carousel-item active">

      <img src="{%static 'internship/images/in2.jpg'%}" class="d-block w-100"
alt="...">

    </div>

    <div class="carousel-item">

      <img src="{%static 'internship/images/home3.jpg'%}" class="d-block w-100"
alt="...">

    </div>

    <div class="carousel-item">

      <img src="{%static 'internship/images/home4.jpg'%}" class="d-block w-100"
alt="...">

    </div>

  </div>

  <button class="carousel-control-prev" type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide="prev">

    <span class="carousel-control-prev-icon" aria-hidden="true"></span>

    <span class="visually-hidden">Previous</span>

  </button>

  <button class="carousel-control-next" type="button" data-bs-
target="#carouselExampleIndicators" data-bs-slide="next">

    <span class="carousel-control-next-icon" aria-hidden="true"></span>

    <span class="visually-hidden">Next</span>

  </button>

</div>

{%include 'internship/common/commonfooter.html'%}


{%include 'internship/common/common_js.html'%}

</body>
```

```
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
    {%load static%}

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
```

```
    <title>login</title>
    {%include 'internship/common/common_css.html'%}
</head>
<body>

    <div class="container-fluid bg-info">

    </div>
    {% include 'internship/common/commonheader.html'%}

    <div class="row  justify-content-center mt-5">
      <div class="col-6" style="border-style: solid;border-
width:3px;border-color: rgb(49, 40, 40);">
    <!-- forms code start -->
    <form method="POSt" action="/login/">
      {%csrf_token%}
        <div class="mb-3">
            <label for="id" class="form-label">provider
id</label>
            <input type="text" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"
name="providerid">
            <div id="emailHelp" class="form-text"></div>
          </div>
        <div class="mb-3">
          <label for="pass" class="form-label">provider
pass</label>
          <input type="password" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"
name="provpass">
          <div id="emailHelp" class="form-text"></div>
        </div>
```

```
        <button type="submit" class="btn btn-
primary">Submit</button>

      </form>

    </div>

   </div>

    <!-- forms code ends -->

<style>

  body{

    background-image: url("{%static
'internship/images/login.jpg'%}");

    background-size: cover;

    background-position: center center;

    background-repeat: no-repeat;

  }

  .form-label{

    color: rgb(206, 208, 44);

  }



</style>


{%include 'internship/common/common_js.html'%}

</body>

</html>
```

About.html

```
<!DOCTYPE html>

{%load static%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    {% include 'internship/common/common_css.html'%}

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    {% include 'internship/common/commonheader.html'%}

    <div class="container-fluid bg-danger">

        <h2 style="text-align: center;">Courses</h2>

        <p style="text-align: center;">This is Courses
page</p>

        </div>

        {% for p in program_data_key%}


        <div class="col">

          <div class="card h-100">

            <!-- <img src="/media/{{e.event_pic}}"
class="card-img-top"

              alt="event pictures" style="width: 100%;height:
100%;" /> -->

            <div class="card-body">

              <h5 class="card-title">Program Name:
{{p.programname}}</h5>

              <h5 class="card-title">Duration:
{{p.duration}}</h5>
```

```
                <h5 class="card-title">Fees description:
{{p.fees}}</h5>

                <!-- <h5 class="card-title">Start
date{{p.startdate}}</h5> -->

                <!-- <h5 class="card-title">End
date{{p.enddate}}</h5> -->

                <h5 class="card-title">Perquisite:
{{p.perquisite}}</h5>

                <h5 class="card-title">Stipend:
{{p.stipend}}</h5>

                <h5 class="card-title">Description:
{{p.description}}</h5>


                <p class="card-text">


                </p>

              </div>

            </div>

          </div>

          {%endfor%}

        </div>

        {%include 'internship/common/commonfooter.html'%}

      {%include 'internship/common/common_js.html'%}

</body>

</html>
```

Contactus.html

```
<!DOCTYPE html>
{%load static%}
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% include 'internship/common/common_css.html'%}
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    {% include 'internship/common/commonheader.html'%}
    {%if messages%}
    {%for m in messages%}
    <div class="alert alert-warning alert-dismissible fade show" role="alert">
        <strong>{{m}}</strong>
        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
    </div>
  {%endfor%}
  {%endif%}




    <form method="post" action="/contactus/">
        {%csrf_token%}


    <div class="row  justify-content-center mt-2">
```

```
      <div class="col-6" style="border-style: solid;border-width:3px;border-color:
rgb(49, 40, 40);">


   <div class="row">




      <div class="mb-0">

      <label for="name" class="from-label">your name</label><i class="fas fa-
home"></i>

      <input type="text" name="username" placeholder="enter your name"
class="form-control required">

      </div>

      <div class="mb-0">

         <label for="email" class="from-label">your email</label>

         <input type="email" name="useremail" placeholder="enter your email"
class="form-control">

      </div>

      <div class="mb-0">

         <label for="phone" class="from-label">your phone</label>

         <input type="number" name="userphone" placeholder="enter your
phonenumber" class="form-control">

      </div>

      <div class="mb-0">

         <label for="query" class="from-label">your query</label>

         <textarea name="userquery" placeholder="enter your query" class="form-
control"></textarea>

      </div>

      <!-- <div>

      <label for="exampleFormControlInput1" class="form-label">Date</label>

        <input type="date" class="form-control" placeholder="program start date"
name="userdate">

      </div> -->
```

```
        <div class="text-center">
            <button class="btn btn-primary">submit</button>
        </div>
        </div>
        </div>
    </form>
    <style>
    body{
        background-image: url("{%static 'internship/images/con1.jpg'%}");
        background-size: cover;
        background-position:top;
        background-repeat: no-repeat;
      }
      .form-control{
        color: rgb(78, 28, 53);
      }

    </style>

    {%include 'internship/common/common_js.html'%}
</body>
</html>
```

Login.html

```
<!DOCTYPE html>

<html lang="en">

    {%load static%}


<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>login</title>

    {%include 'internship/common/common_css.html'%}

</head>

<body>


    <div class="container-fluid bg-info">


    </div>

    {% include 'internship/common/commonheader.html'%}


    <div class="row  justify-content-center mt-5">

      <div class="col-6" style="border-style: solid;border-
width:3px;border-color: rgb(49, 40, 40);">

    <!-- forms code start -->

    <form method="POSt" action="/login/">

      {%csrf_token%}

        <div class="mb-3">

            <label for="id" class="form-label">provider
id</label>

            <input type="text" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"
name="providerid">
```

```
            <div id="emailHelp" class="form-text"></div>
          </div>
        <div class="mb-3">
          <label for="pass" class="form-label">provider
pass</label>
          <input type="password" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"
name="provpass">
          <div id="emailHelp" class="form-text"></div>
        </div>
        <button type="submit" class="btn btn-
primary">Submit</button>
      </form>
    </div>
    </div>
    <!-- forms code ends -->
<style>
  body{
    background-image: url("{%static
'internship/images/login.jpg'%}");
    background-size: cover;
    background-position: center center;
    background-repeat: no-repeat;
  }
  .form-label{
    color: rgb(206, 208, 44);
  }
</style>

{%include 'internship/common/common_js.html'%}
</body>
```

```
</html>
```

Notice.html

```
<!DOCTYPE html>

{%load static%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    {% include 'internship/common/common_css.html'%}

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    {% include 'internship/common/commonheader.html'%}

    <div class="row row-cols-1 row-cols-md-3 g-4">

        {% for n in notice_key%}


          <div class="col">

            <div class="card h-100">

              <!-- <img src="/media/{{e.event_pic}}"
class="card-img-top"

                alt="event pictures" style="width: 100%;height:
100%;" /> -->

              <div class="card-body">

                <h5 class="card-title">Event
Name :{{n.providerid}}</h5>

                <h5 class="card-title">venue
name :{{n.noticetopic}}</h5>

                <h5 class="card-title">event
description :{{n.noticecontents}}</h5>
```

```
                <!-- <h5 class="card-title">event
date{{e.event_date}}</h5> -->
                <p class="card-text">


                </p>
              </div>
            </div>
          </div>
          {%endfor%}
        </div>
    <!-- <h1>this is notice page</h1> -->
    {%include 'internship/common/commonfooter.html'%}
    {%include 'internship/common/common_js.html'%}
</body>
</html>
```

Providerbase.html

```
<!DOCTYPE html>
{%load static%}
{% include 'internship/common/provider_common.html'%}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    {% include 'internship/common/common_css.html'%}
```

```
</head>
<body>
    <!-- card code started -->
    <div class="row row-cols-1 row-cols-md-3 g-4">
        <div class="col">
          <div class="card h-100">
            <img src="{%static
'internship/images/pre.1.jpg'%}" class="card-img-top"
              alt="Hollywood Sign on The Hill" />
            <div class="card-body">
              <h5 class="card-title">Precursor</h5>
              <p class="card-text">
                Covers core java fundamentals after learning
this course; student can become a desktop application
developer. Advance Java.....
              </p>
            </div>
          </div>
        </div>
        <div class="col">
          <div class="card h-100">
            <img src="{%static 'internship/images/info.jpg'%}"
class="card-img-top"
              alt="Palm Springs Road" />
            <div class="card-body">
              <h5 class="card-title">Infosys</h5>
              <p class="card-text">Infosys offers flagship
learning programs bringing digital & life skills for students
and lifelong learners to empower the people. Start learning
for free!</p>
            </div>
          </div>
```

```
        </div>
        <div class="col">
          <div class="card h-100">
            <img src="{%static 'internship/images/pw.jpg'%}"
class="card-img-top"
              alt="Los Angeles Skyscrapers" />
            <div class="card-body">
              <h5 class="card-title">Physics Wallah</h5>
              <p class="card-text">Physics Wallah is India's
top online ed-tech platform that provides affordable and
comprehensive learning experience to students of classes 6 to
12
                </p>
            </div>
          </div>
        </div>

    <!-- card code ends -->

    {%include 'internship/common/commonfooter.html'%}

{%include 'internship/common/common_js.html'%}
</body>
</html>
```

Provideredit_profile.html

```
<!DOCTYPE html>

{%load static%}

{% include 'internship/common/provider_common.html'%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    {% include 'internship/common/common_css.html'%}

    <title>Document</title>

</head>

<body>

    <form method="post" action="/providereditprofile/">

        {%csrf_token%}

        <div class="card" style="width: 50%;height:
auto;background-color: beige;max-width: left;">

            <!-- <img class="card-img-top"
src="/media/{{student_data.student_pic}}" alt="Card image
cap"> -->

            <div class="card-body">

                <h2>your profile Details</h2>

                <p class="card-
text">Name:{{provider_data.organizationname}}

                </p>

                <p class="card-
text">Duration:{{provider_data.ownername}}


                </p>

                <p class="card-text">Email:

                    <input type="email" name="provideremail"
class="form-control"
```

```
            required value={{student_data.email}}></p>


             <p class="card-text">Phone:
                <input type="number" name="providernumber"
class="form-control"
            required value={{student_data.email}}></p>


         <p class="card-text">Address:
            <input type="text" name="Adrress" class="form-
control"
            required value={{student_data.address}}></p>


     <p class="card-text">City:
        <input type="text" name="city" class="form-control"
            required value={{student_data.city}}></p>

     <p class="card-text">Domain:
        <input type="text" name="domain" class="form-control"
            required value={{student_data.domain}}></p>




            </div>
        </div>
```

```
        </div>
    </div>
<!--
    <p>This is provider view profile page</p> -->
    {%include 'internship/common/commonfooter.html'%}
{%include 'internship/common/common_js.html'%}
</body>
</html>
```

Provider_registration.html

```
<!DOCTYPE html>

{%load static%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    {%include 'internship/common/common_css.html'%}

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">


    <title>Document</title>

</head>

<body>

  {%include 'internship/common/commonheader.html'%}




    <div class="container-fluid bg-danger">

    <h5 style="text-align: center;">REGISTRATION</h5>


    </div>

    <div class="row  justify-content-center mt-1" style="color:
bro">

        <div class="col-8" style="border-style: solid;border-
width:3px;border-color: rgb(179, 176, 18);color: aqua;">

            <div class="mb-3">


    <!-- form code start -->

    <form method="POST" action="/registration/">

        {%csrf_token%}
```

```
        <label for="exampleFormControlInput1" class="form-
label">Provider Id</label>

        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider Id" name="providerid">

        <label for="exampleFormControlInput1" class="form-
label">Provider Password</label>

        <input type="password" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider Password" name="provpass">

        <label for="exampleFormControlInput1" class="form-
label">Organization Name</label>

        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="Organization Name"
name="organizationname">

        <label for="exampleFormControlInput1" class="form-
label">Owner Name</label>

        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="please enter Owner
Name" name="ownername">

        <label for="exampleFormControlInput1" class="form-
label">Email</label>

        <input type="email" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider Email" name="email">

        <label for="exampleFormControlInput1" class="form-
label">Phone Number</label>

        <input type="number" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider PhoneNumber" name="phonenumber">

        <label for="exampleFormControlInput1" class="form-
label">Address</label>

        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider Address" name="address">

        <label for="exampleFormControlInput1" class="form-
label">City</label>
```

```
        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider City" name="city">

        <label for="exampleFormControlInput1" class="form-
label">Domain</label>

        <input type="text" class="form-control"
id="exampleFormControlInput1" placeholder="please enter
provider Domain" name="domain">



      </div>

      <div class="mb-3">

        <label for="exampleFormControlTextarea1" class="form-
label">About Organization</label>

        <textarea class="form-control"
id="exampleFormControlTextarea1" name="aboutorganisation"
rows="3"></textarea>

      </div>

      <button class="btn btn-success">Submit</button>

    </form>

    </div>

    </div>

    </div>

    <!-- form code ends -->

    <style>

      body{

        background-image: url("{%static
'internship/images/regis.jpeg'%}");

        background-size: cover;

        background-position: center center;

        background-repeat: no-repeat;

      }

      .form-label{

        color: rgb(206, 208, 44);
```

```
        }
        .form-control{
          background-color:transparent;
        }


    </style>
    {%include 'internship/common/commonfooter.html'%}

    {%include 'internship/common/common_js.html'%}
</body>
</html>
```

Providerviews_course.html

```
<!DOCTYPE html>

{%load static%}

{% include 'internship/common/provider_common.html'%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    {% include 'internship/common/common_css.html'%}

    <title>Document</title>

</head>

<body>

    <form method="post" action="/providercourse/">

    <div class="card" style="width: 50%;height:
auto;background-color: rgb(232, 232, 207);max-width: left;">

        <!-- <img class="card-img-top"
src="/media/{{student_data.student_pic}}" alt="Card image
cap"> -->

       <div class="card-body">

        {% for d in program_data_key%}

           <h2>your Course Details</h2>

           <p class="card-text">Name:{{d.programname}}

           </p>

           <p class="card-text">Duration:{{d.duration}}


           </p>

           <p class="card-number">Email:{{d.fees}}

           </p>

       </p>

       <p class="card-date">StartDate:{{d.startdate}}
```

```
        </p>
    </p>
    <p class="card-date">EndDate:{{d.enddate}}
    </p>
</p>
<p class="card-text">Perquisite:{{d.perquisite}}
</p>
</p>

<p class="card-text">Stipend:{{d.stipend}}
</p>
<p class="card-text">Description:{{d.description}}
</p>



        </div>
    </div>
    {%endfor%}

    <p>This is course view page</p>
    {%include 'internship/common/commonfooter.html'%}
{%include 'internship/common/common_js.html'%}
</body>
</html>
```

Provider.html

```
{% extends
'internship/internship_provider/provider_base.html'%}

{%block title%}

this is provider HOME PAGE

{%endblock%}

{%block style%}

body{background-color:cyan}

p{color:orange}

{%endblock%}


{%block content%}


</div>



    {%if messages%}

    {%for m in messages%}

    <form method="POSt">

        {%csrf_token%}


    <div class="alert alert-warning alert-dismissible fade
show" role="alert">

        <strong>{{m}}</strong>

        <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button>


      </div>

    {%endfor%}
```

```
    {%endif%}
<div class="container-fluid bg-danger">
<h2 style="text-align: center;">Student Home Page</h2>
<p style="text-align: center;">WELCOME TO Course Provider
Page</p>
</div>
<div class="card" style="width: 50%;height: auto;background-
color: beige;max-width: left;">
    <!-- <img class="card-img-top"
src="/media/{{student_data.student_pic}}" alt="Card image
cap"> -->
    <div class="card-body">
        <h2>your profile Details</h2>
        <p class="card-text">Name:{{student_data.name}}
        </p>
        <p class="card-text">Email:{{student_data.email}}
        </p>
        </p>
        <p class="card-text">Phone:{{student_data.phone}}
        </p>
    </div>
</div>
{%endblock%}
```

Providercourse.html

```
<!DOCTYPE html>

{%load static%}

{% include 'internship/common/provider_common.html'%}

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1.0">

    {% include 'internship/common/common_css.html'%}

    <title>Document</title>

</head>

<body>

    <div class="row  justify-content-center mt-2">

        <div class="col-6" style="border-style: solid;border-
width:3px;border-color: rgb(49, 40, 40);">

        <!-- form code start -->

        <form method="post" action="/providercourse/">

            {%csrf_token%}


            <label for="exampleFormControlInput1" class="form-
label">Program Name</label>

            <input type="text" class="form-
control"  placeholder="please enter program Name"
name="program_name">

            <label for="exampleFormControlInput1" class="form-
label">Program Duration</label>

            <input type="text" class="form-control"
placeholder="please enter program duration"
name="program_duration">

            <label for="exampleFormControlInput1" class="form-
label">Program Fees</label>
```

```
            <input type="number" class="form-control"
placeholder="Organization Name" name="program_fees">

            <label for="exampleFormControlInput1" class="form-
label">Program Start</label>

            <input type="date" class="form-control"
placeholder="program start date" name="start_date">

            <label for="exampleFormControlInput1" class="form-
label">Program Ends</label>

            <input type="date" class="form-control"
placeholder="program end date" name="end_date">

            <label for="exampleFormControlInput1" class="form-
label">Program Perquisite</label>

            <input type="text" class="form-control"
placeholder="please enter provider PhoneNumber"
name="program_per">

            <label for="exampleFormControlInput1" class="form-
label">Program Stipend</label>

            <input type="text" class="form-control"
placeholder="please enter provider Address"
name="program_sti">


        <div class="mb-3">

          <label for="exampleFormControlTextarea1"
class="form-label">Program Description</label>

          <textarea class="form-control"
id="exampleFormControlTextarea1" name="program_des"
rows="3"></textarea>




          <button class="btn btn-success">Submit</button>

        </form>

        </div>
```

```
        {%include 'internship/common/commonfooter.html'%}
{%include 'internship/common/common_js.html'%}
</body>
</html>
```

Providerprofile.html

```
<!DOCTYPE html>
{%load static%}
{% include 'internship/common/provider_common.html'%}
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    {% include 'internship/common/common_css.html'%}
    <title>Document</title>
</head>
<body>
    <form method="post" action="/registration/">

    <div class="card" style="width: 50%;height:
auto;background-color: beige;max-width: left;">
         <!-- <img class="card-img-top"
src="/media/{{student_data.student_pic}}" alt="Card image
cap"> -->
        <div class="card-body">
            <h2>your profile Details</h2>
            <p class="card-
text">Name:{{provider_data.organizationname}}
            </p>
```

```
            <p class="card-
text">Duration:{{provider_data.ownername}}


            </p>
            <p class="card-text">Email:{{provider_data.email}}
            </p>
        </p>
        <p class="card-
text">Phone:{{provider_data.phonenumber}}
        </p>
    </p>
    <p class="card-text">Address:{{provider_data.address}}
    </p>
</p>
<p class="card-text">City:{{provider_data.city}}
</p>
</p>
<p class="card-text">Domain:{{provider_data.domain}}
</p>


        </div>
     </div>



    <p>This is provider view profile page</p>
    {%include 'internship/common/commonfooter.html'%}
{%include 'internship/common/common_js.html'%}
</body>
</html>
```

Commoncss.html

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bo
otstrap.min.css" rel="stylesheet" integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQI
u9" crossorigin="anonymous">

<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.min.css" integrity="sha512-
q3eWabyZPc1XTCmF+8/LuE1ozpg5xxn7iO89yfSOd5/oKvyqLngoNGsx8jq92Y
8eXJ/IRxQbEC+FGSYxtk2oiw==" crossorigin="anonymous"
referrerpolicy="no-referrer" />
```

Commonjs.html

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/boot
strap.bundle.min.js" integrity="sha384-
HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgr
km" crossorigin="anonymous"></script>
```

Commonfooter.html

```
<!-- footer code start -->
<footer class="bg-light text-center text-white">
    <!-- Grid container -->
    <div class="container p-4 pb-0">
      <!-- Section: Social media -->
      <section class="mb-4">
        <!-- Facebook -->
        <a
          class="btn text-white btn-floating m-1"
          style="background-color: #3b5998;"
          href="#!"
          role="button"
          ><i class="fab fa-facebook-f"></i
        ></a>

        <!-- Twitter -->
        <a
          class="btn text-white btn-floating m-1"
          style="background-color: #55acee;"
          href="#!"
          role="button"
          ><i class="fab fa-twitter"></i
        ></a>

        <!-- Google -->
        <a
          class="btn text-white btn-floating m-1"
          style="background-color: #dd4b39;"
```

```
      href="#!"
      role="button"
      ><i class="fab fa-google"></i
    ></a>

    <!-- Instagram -->
    <a
      class="btn text-white btn-floating m-1"
      style="background-color: #ac2bac;"
      href="#!"
      role="button"
      ><i class="fab fa-instagram"></i
    ></a>

    <!-- Linkedin -->
    <a
      class="btn text-white btn-floating m-1"
      style="background-color: #0082ca;"
      href="#!"
      role="button"
      ><i class="fab fa-linkedin-in"></i
    ></a>


  </section>
  <!-- Section: Social media -->
</div>
<!-- Grid container -->
```

```
    <!-- Copyright -->

    <div class="text-center p-3" style="color:
red;"background-color: rgba(0, 0, 0, 0.2);">

      © 2024 Copyright  Amit Dubey:


    </div>

    <!-- Copyright -->

  </footer>

<!-- footer code ends -->
```

Commonheader.html

```
<style>
    .nav-link{
      color:rgb(187, 181, 66);
      font-size: 20px;
      font-family: 'times roman';
    }

    .dropdown-item{
      color: rgb(172, 39, 21);
      font-size: 20px;
      font-family: 'times roman';
    }
    a:hover{
      color:rgb(150, 220, 236)!important
    }
    .navbar-brand{
      color:rgb(75, 172, 19);
    }
    </style>

    <nav class="navbar navbar-expand-lg bg-dark">
        <div class="container">
          <a class="navbar-brand"
href="https://in.indeed.com/"><U><b>INTERNSHIP HUB</b></U></a>
          <button class="navbar-toggler" type="button" data-
bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
```

```
          <div class="collapse navbar-collapse"
id="navbarSupportedContent">

            <ul class="navbar-nav me-auto mb-2 mb-lg-0">

              <li class="nav-item">

                <a class="nav-link" aria-current="page"
href="{{'/'}}">Home</a>

              </li>

              <li class="nav-item">

                <a class="nav-link"
href="{{'/about/'}}">Courses</a>

              </li>

              <li class="nav-item">

                <a class="nav-link"
href="{{'/feedback/'}}">Give Feedback</a>

              </li>


              <li class="nav-item">

                <a class="nav-link"
href="{{'/contactus/'}}">Contact Us</a>

                </li>

                <li class="nav-item">

                  <a class="nav-link"
href="{{'/notice/'}}">Notice</a>

                  </a>

                  </li>


                <li class="nav-item">

                  <a class="nav-link"
href="{{'/login/'}}">Login</a>

                  </a>

                  </li>
```

```
                <li class="nav-item">
                  <a class="nav-link"
href="{{'/registration/'}}">Registration</a>
                  </a>
                  </li>

              <!-- <li class="nav-item dropdown">
                <a class="nav-link dropdown-toggle "
href="#" role="button" data-bs-toggle="dropdown" aria-
expanded="false">
                  Profile
                </a>
                <ul class="dropdown-menu">
                  <li><a class="dropdown-item"
href="#"><h5>Login as Student</h5></a></li>
                  <li><hr class="dropdown-divider"></li>
                  <li><a class="dropdown-item"
href="{{'/login/'}}">Login as Internship Provider</a></li>

                </ul>
              </li> -->

            </ul>
            <form class="d-flex" role="search">
              <input class="form-control me-2" type="search"
placeholder="Search" aria-label="Search">
              <button class="btn btn-outline-success"
type="submit">Search</button>
            </form>
          </div>
        </div>
      </nav>
```

Providercommon.html

```
<style>
  .nav-link{
      color:rgb(187, 181, 66);
      font-size: 20px;
      font-family: 'times roman';
    }
    a:hover{
      color:rgb(150, 220, 236)!important
    }




</style>
<div class="container-fluid bg-danger">


  <h5 style="text-align: center;margin: 0%;">Provider
Dashboard</h5>


  </div>
<nav class="navbar navbar-expand-lg bg-warning">
    <div class="container-fluid">




      <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
```

```
      <div class="collapse navbar-collapse"
id="navbarSupportedContent">

        <ul class="navbar-nav me-auto mb-0 mb-lg-0">




          <li class="nav-item dropdown">

            <a class="nav-link dropdown-toggle" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false">

              Profile

            </a>

            <ul class="dropdown-menu">

              <li><a class="dropdown-item"
href="{{'/providerviewprofile/'}}">View Profile</a></li>

              <li><a class="dropdown-item"
href="{{'/provideredituprofile/'}}">Edit Profile</a></li>

              <!-- <li><a class="dropdown-item"
href="{{'/providercourse/'}}">Add Course</a></li>

              <li><a class="dropdown-item"
href="{{'/providerviewcourse/'}}">View Course</a></li> -->



            </ul>

          </li>




          <li class="nav-item dropdown">

            <a class="nav-link dropdown-toggle" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false">

              courses

            </a>

            <ul class="dropdown-menu">
```

```
              <!-- <li><a class="dropdown-item"
href="{{'/providerviewprofile/'}}">View Profile</a></li>

              <li><a class="dropdown-item"
href="{{'/provideredituprofile/'}}">Edit Profile</a></li> -->

              <li><a class="dropdown-item"
href="{{'/providercourse/'}}">Add Course</a></li>

              <li><a class="dropdown-item"
href="{{'/providerviewcourse/'}}">View Course</a></li>


            </ul>


          <li class="nav-item">

            <a class="nav-link"
href="{{'/login/'}}">Logout</a>




        </ul>


      </div>

    </div>

  </nav>
```

Style.css

```
body{
    background: url('images/pic1.jpg') no-repeat center
center/cover;


}
/* login form */
.form{
    display: flex;
    flex-direction: column;
    height: 500px;
    width: 400px;
    /* border: 1px solid black; */
    align-items: center;
    margin: auto;
    margin-top: 50px ;
    background-color: rgba(0,0, 0, 0.5);
    box-shadow: inset -5px -5px rgba(0,0,0.5);
    border-radius: 25px;
}
.form h1{
    color: white;
    font-size: 2rem;
    border-bottom: 4px solid rgba(255,255,255,0.5);
    margin: 50px;
}
/* Boxes */
.box{
    padding:12px ;
    margin: 20px;
```

```
    width: 65%;
    border: none;
    outline: none;
    border-radius: 20px;
    background-color: rgba(0, 0, 0 , 0.5);
    box-shadow: inset -3px -3px rgba(0, 0, 0, 0.5)
    color(white)
}
#submit{
    padding:10px 20px ;
    margin-top: 50px;
    width: 100px;
    background-color: rgba(0,0,0, 0.5);
    box-shadow: inset -3px -3x rgba(0, 0, 0, 0.5);
    color: white;
    border: none;
    outline: none;
    border-radius: 20px;
    font-size: 1rem;
}

#submit:hover{
    cursor: pointer;
    background-color: rgba(255, 255,255, 0.1);
    color: white;

}
```

Backend-:

admin.py

```
from django.contrib import admin

from .models import Provider, ProgramDetails, Contactus,
Notice


# Register your models here.

admin.site.register(Provider)

admin.site.register(ProgramDetails)

admin.site.register(Contactus)

admin.site.register(Notice)
```

models.py

```
from django.db import models
from django.utils import timezone


# Create your models here.


class Provider(models.Model):
    providerid = models.CharField(max_length=45,
primary_key=True)
    provpass = models.CharField(max_length=45)
    organizationname = models.CharField(max_length=100)
    ownername = models.CharField(max_length=100)
    email = models.EmailField(max_length=45)
    phonenumber = models.CharField(max_length=10)
    address = models.CharField(max_length=100)
    city = models.CharField(max_length=50)
    domain = models.CharField(max_length=100)
    aboutorganisation = models.TextField()


    def _str_(self):
        return self.ownername


class ProgramDetails(models.Model):
    Providerid = models.ForeignKey(Provider,
on_delete=models.DO_NOTHING)
    programname = models.CharField(max_length=100)
    duration = models.CharField(max_length=20)
    fees = models.CharField(max_length=100)
    startdate = models.DateField(default=timezone.now)
```

```
    enddate = models.DateField(default=timezone.now)
    perquisite = models.CharField(max_length=255)
    stipend = models.CharField(max_length=3)
    description = models.CharField(max_length=255)

    def _str_(self):
        return self.programname

class Feedback(models.Model):
    name = models.CharField(max_length=100)
    email = models.CharField(max_length=20)
    date = models.DateField(default=timezone.now)
    feedbacktext = models.TextField()
    rating = models.IntegerField()

    def _str_(self):
        return self.name

class Contactus(models.Model):
    name = models.CharField(max_length=45)
    email = models.EmailField(max_length=45)
    phone = models.CharField(max_length=10)
    question = models.TextField()
    date = models.DateField(default=timezone.now)

    def _str_(self):
        return self.name
```

```
class Notice(models.Model):
    providerid = models.CharField(max_length=45)
    noticetopic = models.CharField(max_length=100)
    noticecontents = models.CharField(max_length=255)
    date = models.DateField(default=timezone.now)

    def _str_(self):
        return self.noticetopic

# class Registartion(models.Model):
```

Views.py

```
from django.shortcuts import render,HttpResponse

from .models import
Provider,Feedback,Contactus,Notice,ProgramDetails

from django.contrib import messages

def home(request):

    return render(request,"internship/html/home.html")


def about_internship(request):

    program_list=ProgramDetails.objects.all()

    program_contents={

        "program_data_key":program_list,

        "title":"MyHomePage"

    }

    return
render(request,"internship/html/about_internship.html",program
_contents)


def feedback(request):

    if request.method=="GET":


        return render(request,"internship/html/feedback.html")

    if request.method=="POST":

        fn=request.POST["name"]

        fe=request.POST["email"]

        fd=request.POST["date"]

        ft=request.POST["feedbacktext"]

        fr=request.POST["rating"]

        feedback_obj=Feedback(name=fn,email=fe,date=fd,feedbac
ktext=ft,rating=fr)
```

```
        feedback_obj.save()
        messages.success(request,"Thanks For Your Feedback")
        return render(request,"internship/html/feedback.html")


def contactus(request):
    if request.method=="GET":
        return render(request,"internship/html/contactus.html")
    if request.method=="POST":
        un=request.POST["username"]
        ue=request.POST["useremail"]
        up=request.POST["userphone"]
        uq=request.POST["userquery"]
        # ud=request.post["userdate"]
        contactus_obj=Contactus(name=un,email=ue,phone=up,ques
tion=uq)
        contactus_obj.save()
        messages.success(request,"your query is resolve
shortly")
        return render(request,"internship/html/home.html")



# def needhelp(request):
#     return render(request,"internship/html/needhelp.html")
#     return render(request,"internship/html/needhelp.html")


# def login(request):
#     return render(request,"internship/html/login.html")


def registration(request):
    if request.method == "GET":
```

```
        return
render(request,"internship/internship_provider/provider_regist
ration.html")


    if request.method == "POST":

        p_id=request.POST["providerid"]

        p_pass=request.POST["provpass"]

        o_name=request.POST["organizationname"]

        own_name=request.POST["ownername"]

        o_email=request.POST["email"]

        o_phone=request.POST["phonenumber"]

        o_add=request.POST["address"]

        o_city=request.POST["city"]

        o_domain=request.POST["domain"]

        o_about=request.POST["aboutorganisation"]

        print(p_id,p_pass)

        registration_obj=Provider(providerid=p_id,provpass=p_p
ass,organizationname=o_name,ownername=own_name,email=o_email,p
honenumber=o_phone,address=o_add,city=o_city,domain=o_domain,a
boutorganisation=o_about)

        registration_obj.save()

        return render(request,"internship/html/login.html")

def notice(request):

    notice_object_list=Notice.objects.all()

    notice_contents={

        "notice_key":notice_object_list,

        "title":"MyHomePage"

    }

    return
render(request,"internship/html/notice.html",notice_contents)

# Create your views here.
```

urls.py

```
from django.urls import path,include

from .import views,provider_views

urlpatterns=[

    path("",views.home),

    path("about/",views.about_internship),

    path("feedback/",views.feedback),

    path("contactus/",views.contactus,name="contactus"),

    path("login/",provider_views.login,name="login"),

    path("registration/",views.registration,name="registration
"),

    path("providercourse/",provider_views.providercourse),

    path("providerviewcourse/",provider_views.providerviewcour
se),

    path("providerviewprofile/",provider_views.providerviewpro
file),

    path("provideреditprofile/",provider_views.provideреditpro
file),

    path("providercourse/",provider_views.providercourse),

    path("notice/",views.notice),

]
```

Provider_views.py

```
from django.shortcuts import render,redirect

from .models import Provider,ProgramDetails

from django.contrib import messages



def login(request):

    if request.method=="GET":

        return render(request,"internship/html/login.html")

    if request.method=="POST":

        p_id=request.POST["providerid"]

        p_pass=request.POST.get("provpass")

        print(p_id,p_pass)

        Provider_list=Provider.objects.filter(providerid=p_id,
provpass=p_pass)

        length=len(Provider_list)


        if length>0:

            request.session["provider_key"]=p_id


            #return redirect("login")

            provider_obj=Provider.objects.get(providerid=p_id)

#          print(provider_obj)


#        p_dict={"provider_data":provider_obj}


            return
render(request,'internship/internship_provider/provider.html')

        else:

            return render(request,'internship/html/login.html')
```

```
def providercourse(request):
    if request.method == "GET":
        return
render(request,"internship/internship_provider/providercourse.
html")
    if request.method == "POST":


        p_id= request.session["provider_key"]
        print(p_id)
        provider_obj=Provider.objects.get(providerid=p_id)
        print(provider_obj)
        p_name=request.POST["program_name"]
        p_duration=request.POST["program_duration"]
        p_fees=request.POST["program_fees"]
        # p_sdate=request.POST["start_date"]
        # p_edate=request.POST["end_date"]
        p_perquisite=request.POST["program_per"]
        p_stipend=request.POST["program_sti"]
        # p_description=request.POST["program_des"]
        course_obj=ProgramDetails(Providerid=provider_obj,prog
ramname=p_name,duration=p_duration,fees=p_fees,perquisite=p_pe
rquisite,stipend=p_stipend)
        course_obj.save()
        return
render(request,"internship/internship_provider/providerprofile.
html")
def providerviewcourse(request):
    provider_obj_list=Provider.objects.all()
    course_obj_list=ProgramDetails.objects.all()
    course_contents={
```

```
        "provider_key":provider_obj_list,
        "program_data_key":course_obj_list
    }
    return
render(request,"internship/internship_provider/provider_viewco
urse.html",course_contents)


def providerviewprofile(request):
        p_id= request.session["provider_key"]
        print(p_id)
        provider_obj=Provider.objects.get(providerid=p_id)
        print(provider_obj)
        p_dict={"provider_data":provider_obj}
        return
render(request,"internship/internship_provider/providerprofile.
html",p_dict)


# def providerviewprofile(request):
#     return
render(request,"internship/internship_provider/providerprofile.
html")



    #     employee_object_list=Employee.objects.all()
    # event_object_list=Event.objects.all()
    # employee_contents={
    #     "employee_key":employee_object_list,



    #     "event_key":event_object_list
    # }
```

```
    # return
render(request,"myapp/html/home.html",employee_contents)


def providereditprofile(request):
    if request.method=="GET":
        p_id= request.session["provider_key"]
        print(p_id)
        provider_obj=Provider.objects.get(providerid=p_id)
        print(provider_obj)
        p_dict={"provider_data":provider_obj}
        return
render(request,"internship/internship_provider/provider_editpr
ofile.html",p_dict)
    if request.method=="POST":
        pe=request.POST["provideremail"]
        pp=request.POST["providernumber"]
        pa=request.POST["address"]
        pc=request.POST["city"]
        pd=request.POST["domain"]
        p_id= request.session["provider_key"]
        provider_obj=Provider.objects.get(providerid=p_id)
        provider_obj.email=pe
        provider_obj.phonenumber=pp
        provider_obj.address=pa
        provider_obj.city=pc
        provider_obj.domain=pd
        provider_obj.save()
        p_dict={"provider_data":provider_obj}
        return
render(request,"internship/internship_provider/providerprofile.
html",p_dict)
```

Apps.py

```
from django.apps import AppConfig


class InternshipConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'internship'
```

0001_initial.py

```
# Generated by Django 4.2.3 on 2023-08-19 03:05


from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [

    ]

    operations = [

        migrations.CreateModel(

            name='Provider',

            fields=[

                ('providerid', models.CharField(max_length=45,
primary_key=True, serialize=False)),

                ('provpass', models.CharField(max_length=45)),

                ('organizationname',
models.CharField(max_length=100)),

                ('ownername',
models.CharField(max_length=100)),

                ('email', models.EmailField(max_length=45)),

                ('phonenumber',
models.CharField(max_length=10)),

                ('address', models.CharField(max_length=100)),

                ('city', models.CharField(max_length=50)),

                ('domain', models.CharField(max_length=100)),

                ('aboutorganisation', models.TextField()),

            ],

        ),

    ]
```

0002_Programdetails.py

```
# Generated by Django 4.2.3 on 2023-08-19 07:52


from django.db import migrations, models
import django.db.models.deletion
import django.utils.timezone


class Migration(migrations.Migration):


    dependencies = [
        ('internship', '0001_initial'),
    ]


    operations = [
        migrations.CreateModel(
            name='ProgramDetails',
            fields=[
                ('id', models.BigAutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
                ('programname',
models.CharField(max_length=100)),
                ('duration', models.CharField(max_length=20)),
                ('fees', models.CharField(max_length=100)),
                ('startdate',
models.DateField(default=django.utils.timezone.now)),
                ('enddate',
models.DateField(default=django.utils.timezone.now)),
                ('perquisite',
models.CharField(max_length=255)),
                ('stipend', models.CharField(max_length=3)),
```

```
                ('description',
models.CharField(max_length=255)),
                ('Providerid',
models.ForeignKey(on_delete=django.db.models.deletion.DO_NOTHI
NG, to='internship.provider')),
            ],
        ),
    ]
```

0003_contactus_notice.py

```
# Generated by Django 4.2.3 on 2023-08-19 08:00

from django.db import migrations, models
import django.utils.timezone


class Migration(migrations.Migration):

    dependencies = [
        ('internship', '0002_programdetails'),
    ]

    operations = [
        migrations.CreateModel(
            name='contactus',
            fields=[
                ('id', models.BigAutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
```

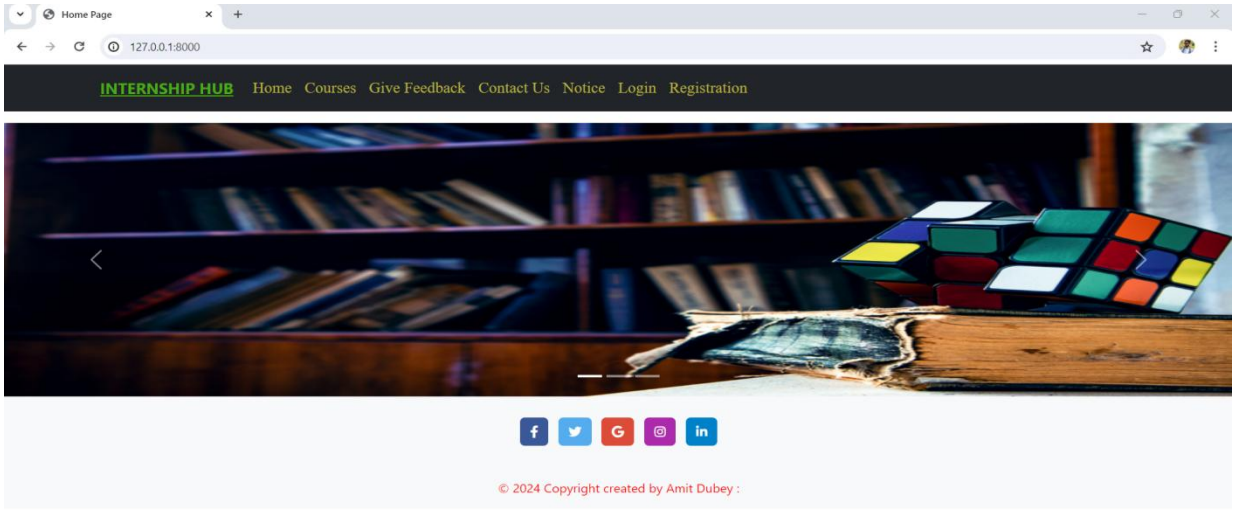
```
                ('name', models.CharField(max_length=45)),
                ('email', models.EmailField(max_length=45)),
                ('phone', models.CharField(max_length=10)),
                ('question', models.TextField()),
                ('date',
models.DateField(default=django.utils.timezone.now)),
            ],
        ),
        migrations.CreateModel(
            name='Notice',
            fields=[
                ('id', models.BigAutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
                ('providerid',
models.CharField(max_length=45)),
                ('noticetopic',
models.CharField(max_length=100)),
                ('noticecontents',
models.CharField(max_length=255)),
                ('date',
models.DateField(default=django.utils.timezone.now)),
            ],
        ),
    ]
```

0004_feedback.py

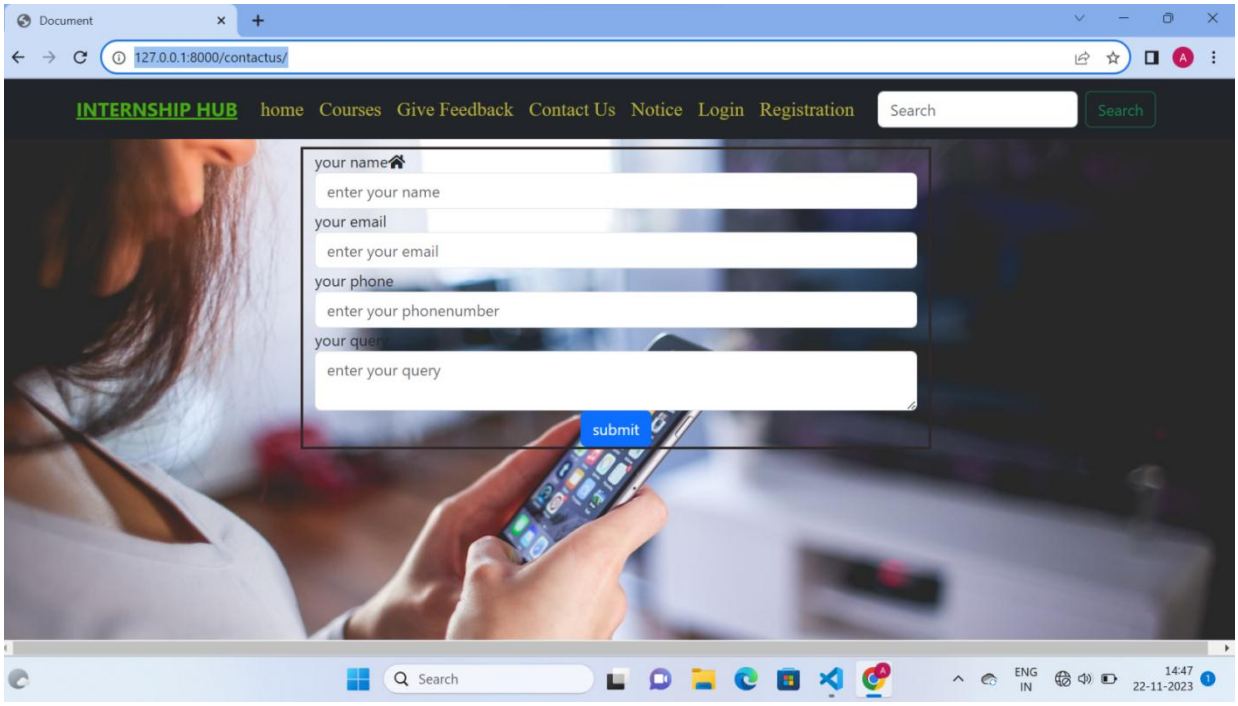
```
# Generated by Django 4.2.3 on 2023-08-19 11:40

from django.db import migrations, models
import django.utils.timezone


class Migration(migrations.Migration):

    dependencies = [
        ('internship', '0003_contactus_notice'),
    ]

    operations = [
        migrations.CreateModel(
            name='Feedback',
            fields=[
                ('id', models.BigAutoField(auto_created=True,
primary_key=True, serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=100)),
                ('email', models.CharField(max_length=20)),
                ('date',
models.DateField(default=django.utils.timezone.now)),
                ('feedbacktext', models.TextField()),
                ('rating', models.IntegerField()),
            ],
        ),
    ]
```

Screenshot:-

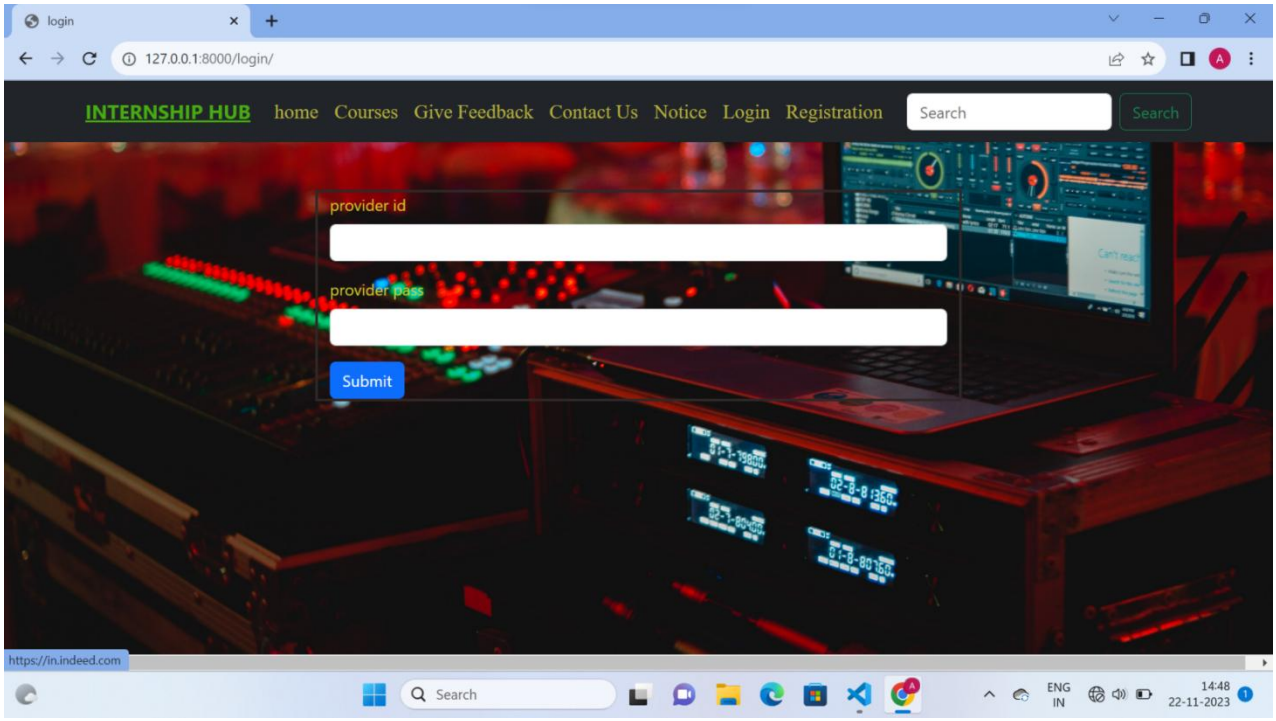
Home



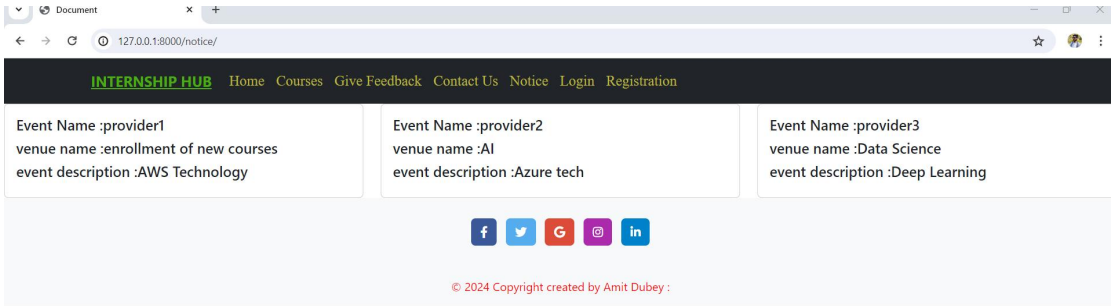
Contactus

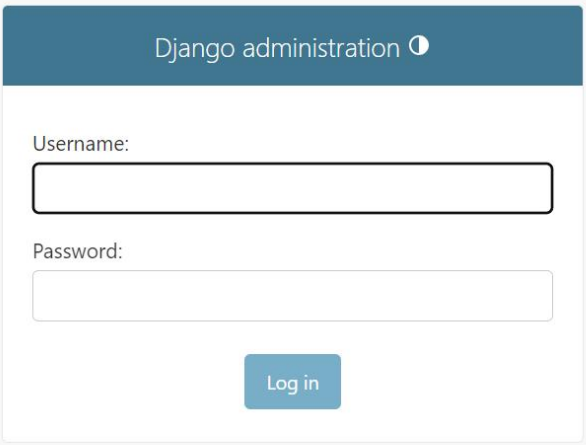


Login

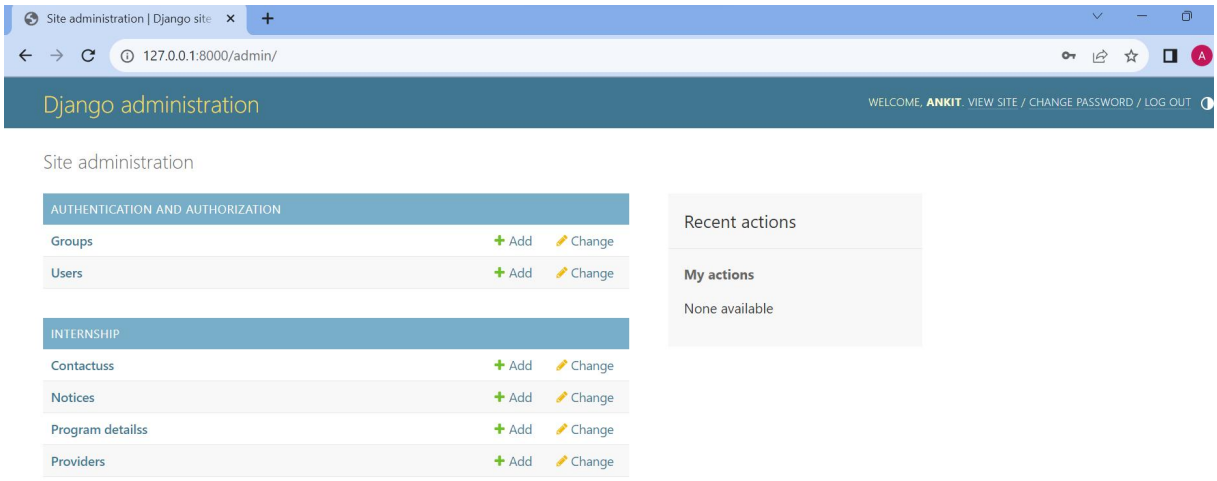


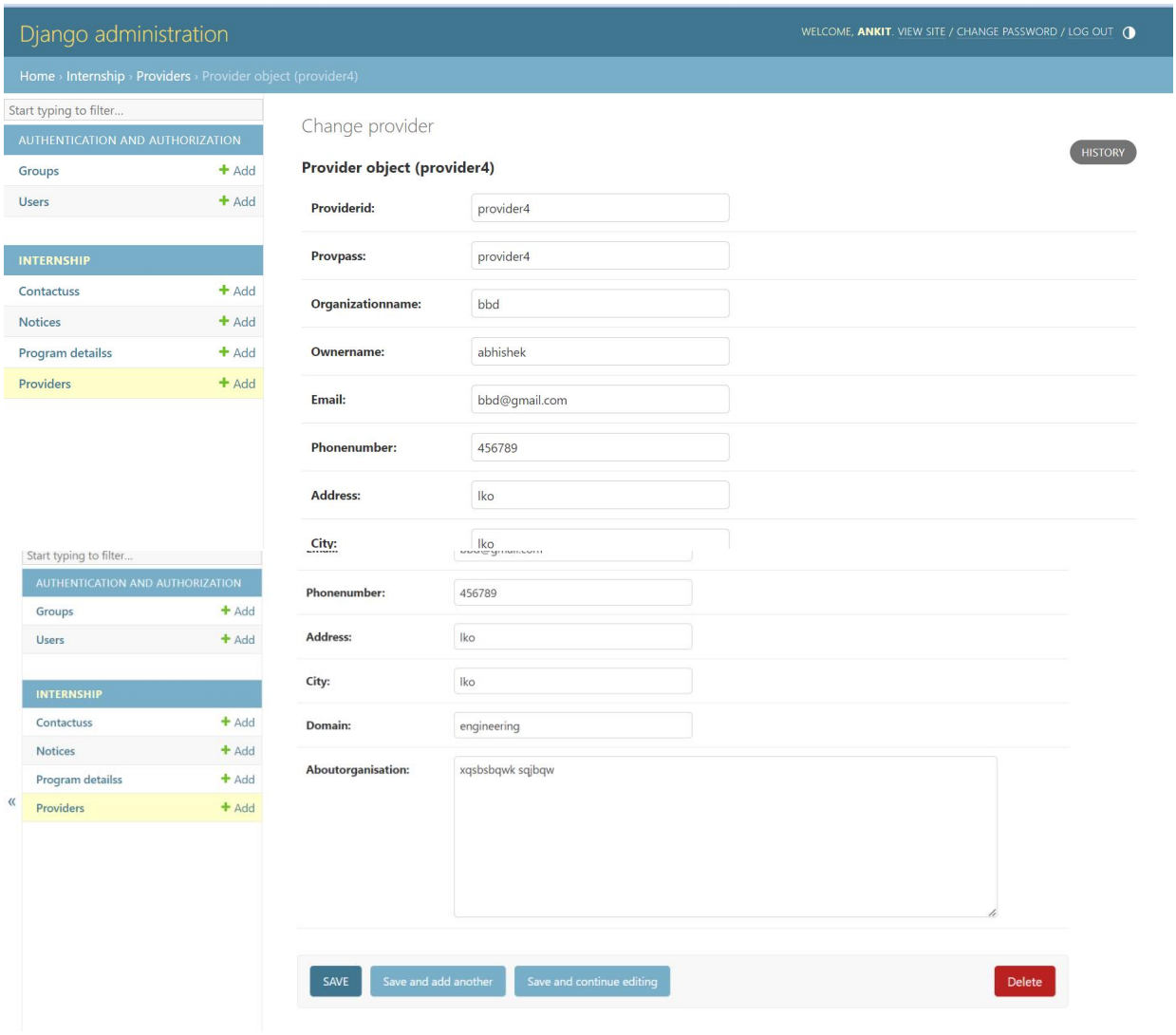
Notice

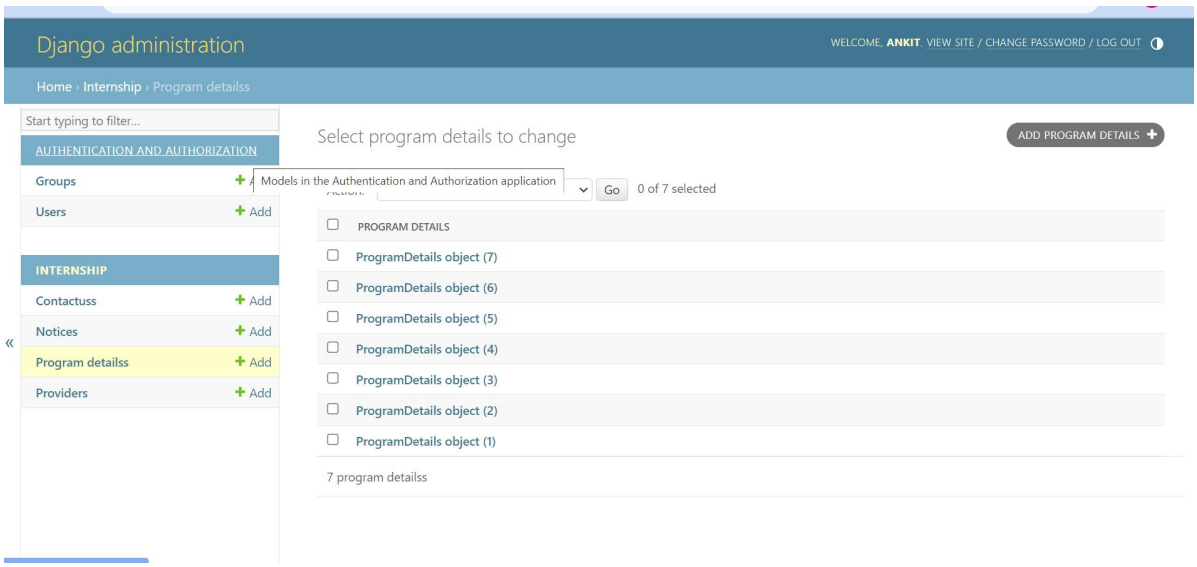



Django administration
Username:
Password:
Log in

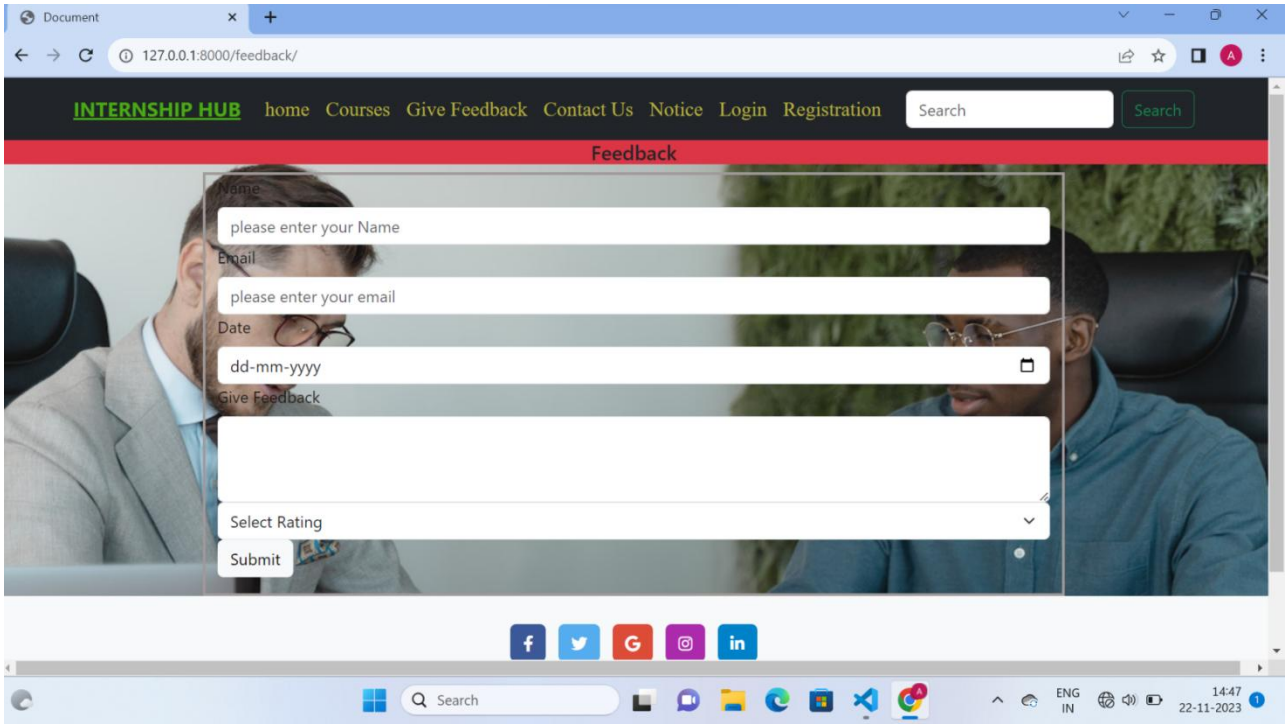
Admin.py


Site administration | Django site
127.0.0.1:8000/admin/
Django administration
WELCOME, ANKIT. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
INTERNSHIP
Contactuss + Add Change
Notices + Add Change
Program detailss + Add Change
Providers + Add Change
Recent actions
My actions
None available


Django administration
WELCOME, ANKIT. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Internship › Providers › Provider object (provider4)
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
INTERNSHIP
Contactuss + Add
Notices + Add
Program detailss + Add
Providers + Add
Change provider
HISTORY
Provider object (provider4)
Providerid: provider4
Provpass: provider4
Organizationname: bbd
Ownername: abhishek
Email: bbd@gmail.com
Phonenumber: 456789
Address: lko
City: lko
Domain: engineering
Aboutorganisation: xqsbsbqwk sqjbqw
SAVE
Save and add another
Save and continue editing
Delete


Django administration
WELCOME, ANKIT. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Internship › Program detailss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Models in the Authentication and Authorization application
Groups + Add
Users + Add
INTERNSHIP
Contactuss + Add
Notices + Add
Program detailss + Add
Providers + Add
Select program details to change
ADD PROGRAM DETAILS +
Go
0 of 7 selected
PROGRAM DETAILS
ProgramDetails object (7)
ProgramDetails object (6)
ProgramDetails object (5)
ProgramDetails object (4)
ProgramDetails object (3)
ProgramDetails object (2)
ProgramDetails object (1)
7 program detailss

Feedback


Document
127.0.0.1:8000/feedback/
INTERNSHIP HUB
home
Courses
Give Feedback
Contact Us
Notice
Login
Registration
Search
Search
Feedback
Name
please enter your Name
Email
please enter your email
Date
dd-mm-yyyy
Give Feedback
Select Rating
Submit
Search
ENG
IN
14:47
22-11-2023

CONCLUSION

The project deserves Ultimate Service Hub web application successfully completed**.**

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project is to develop a web application for the students who searches for internship and for the internship providing companies that provides internships.

This project has given us great satisfaction in having designed an application.

Technology has made significant progress over the years to provide consumers a better online to find internships experience and will continue to do so for years to come. With the rapid growth of students and providing companies.

Future Scope:

- The system can be implemented at any multinational branded Intership providing companies through online portal. Providing outlet chain in the locality. System 24*7 student order accepting facility and recommend providing Internship which can make students happy.
- If the Internship companies are providing an online portal where their Students can enjoy the ease of getting Internship from anywhere.
- By making web application, we can gain Student's trust.

- Can be used by the Students over a small area within the city.

REFERENCES

https://www.precursor.com

https://geeksforgeeks.com

https://www.javatpoint.com

HTML Tutorial (w3schools.com)

CSS Tutorial (w3schools.com)

JavaScript Tutorial (w3schools.com)

Visual Studio Code - Code Editing. Redefined

https://getbootstrap.com/

BIOGRAPHY

	Amit Dubey is a final year Master of Computer Applications (MCA) student at Shri Ramswaroop Memorial University (SRMU). With a passion for technology and a keen interest in computer science, Amit has consistently demonstrated his dedication to academic excellence and innovation throughout his academic journey. Born and raised in Gonda, Amit exhibited an early aptitude for computers and programming, which led him to pursue a career in the field of information technology