PLACEMENT PORTAL FOR GEC WAYANAD

A PROJECT REPORT

submitted by

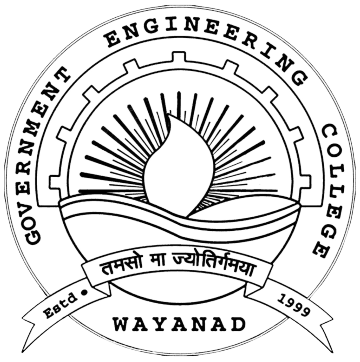
ADITH KOLIYOT (WYD20CS004)
AMAAN ZAIN N (WYD20CS012)
ANU ANTONY (WYD20CS015)
NIVED NARAYANAN KK (WYD20CS045)

to

The APJ Abdul Kalam Technological University
in fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology
in
Computer Science and Engineering



Department of Computer Science and Engineering

Government Engineering College, Wayanad
Thalappuzha – 670644

DECEMBER, 2023

DECLARATION

I, on behalf of authors of the report: Adith Koliyot, Amaan Zain N, Anu Antony, Nived Narayanan K K, hereby declare that the project report "PLACEMENT PORTAL FOR GEC Wayanad " submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Ms. Dhanya Raj P (Asst. Professor, CSE Department). This submission represents our ideas in our own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Thalappuzha
Date: 08-12-2023

Nived Narayanan KK

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE, WAYANAD
THALAPPUZHA – 670644

CERTIFICATE

This is to certify that the report entitled **"PLACEMENT PORTAl FOR GEC WAYANAD"** submitted by **Adith Koliyot (WYD20CS004), Amaan Zain N (WYD20CS012), Anu Antony (WYD20CS015), Nived Narayanan K K (WYD20CS045)** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Supervisor(s)

Department Seal

Project Coordinator(s) Head of the Department

ACKNOWLEDGEMENTS

I am greatly indebted to God Almighty for being the guiding light throughout with his abundant grace and blessings that strengthened us to do this endeavour with confidence. I would like to acknowledge, Prof. Dhanya Raj P, Assistant Professor, Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for inspiring and providing sincere guidance throughout the project. I also express our sincere thanks to the project coordinator Prof. Dhanya Raj P, Assistant Professor, Dept. of Computer Science and Engineering, Prof. Suranya Das M S , Assistant Professor , Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for their valuable suggestions and constant encouragement. I am extremely grateful to Dr. Gilesh M P, Head of the Department, Computer Science and Engineering, Govt. Engineering College, Wayanad for his wise guidance and supportive attitude. I would also like to express our sincere gratitude to each Dept. of Computer Science and Engineering, Govt member. Engineering College, Wayanad, for their kind co-operation and encouragement that helped us in completing the project. I would like to acknowledge Dr. Jasmin E A, Principal of Govt. Engineering College, Wayanad for providing the facilities. I would also like to extend my gratitude to all well-wishers and friends who supported us in presenting the project.

ABSTRACT

In the current competitive job market, college students often face challenges in securing placements due to a lack of proper guidance and scattered resources. This project proposes the development of a placement portal for Government Engineering College, Wayanad (GECW). The portal aims to modernize and automate the conventional placement procedures by integrating all users onto a single platform. Users, including student can benefit from features such as resume building, on-campus and off-campus job applications, interview experiences, placement statistics, and guided paths. The proposed solution addresses limitations in existing procedures, providing real-time updates, efficient communication, and enhanced transparency. This work outlines the system architecture, module descriptions, and the software framework employed. As the development progresses, interim results and a work plan for future phases are discussed. The success of the portal hinges on overcoming challenges related to user engagement, data security, and technology accessibility. By addressing these limitations, the placement portal aspires to offer an effective solution for optimizing the placement process at GECW.

CONTENTS

Contents		Page No.
ACKNOWLEDGEMENTS		i
ABSTRACT		ii
LIST OF TABLES		iv
LIST OF FIGURES		v
CHAPTER 1.	INTRODUCTION	1
	1.1 Problem Description	1
CHAPTER 2.	LITERATURE REVIEW	3
	2.1 Naukri[1]	3
	2.2 Linkedin[2]	5
	2.3 Geeks for geeks[5]	7
	2.4 Glassdoor[6]	9
	2.5 Placement india[7]	11
	2.6 Summary of existing works	12
	2.7 Motivation for the project proposal	15
	2.8 Problem Statement	15
CHAPTER 3.	REQUIREMENT ANALYSIS	16
	3.1 Introduction	16
	3.1.1 Purpose	16
	3.1.2 Definitions and Acronyms	17
	3.2 Overall Description	17
	3.2.1 User Needs	18
	3.2.2 Assumptions and Dependencies	18
	3.3 Method of Requirement Elicitation	19
	3.3.1 Brainstorming	19
	3.3.2 System Study	20
	3.4 System Features and Requirements	20
	3.4.1 Functional Requirements	20
	3.4.2 External Interface Requirements	26
	3.4.3 Nonfunctional Requirements	27
	3.4.4 Hardware Requirements	28
	3.4.5 Software Requirements	28
	3.5 Process Model	29
	3.6 Feasibility Study	31
	3.6.1 Economic Feasibility	31
	3.6.2 Technical Feasibility	31
	3.6.3 Operational Feasibility	32

3.6.4 Scheduling Feasibility 32
CHAPTER 4. ARCHITECTURAL DESIGN 33
4.1 Module Description 35
CHAPTER 5. DESIGN AND IMPLEMENTATION 37
5.1 Introduction 37
5.2 Data Flow Diagrams 37
5.2.1 DFD level 0 37
5.2.2 DFD level 1 39
5.2.3 DFD level 2 41
5.3 Work Plan and Task Allocation 42
CHAPTER 6. CODING 44
6.1 Introduction 44
6.1.1 Frameworks 45
6.1.2 Tools 46
6.1.3 Library 47
6.2 Sample Codes 54
6.3 Summary 60
CHAPTER 7. RESULTS AND DISCUSSION 61
7.1 Introduction 61
7.2 Results 63
CHAPTER 8. DOCUMENTATION 72
8.1 Introduction 72
8.2 Working with the Product 73
8.2.1 User characteristics 73
8.2.2 Product functions 73
8.3 Contact 74
CHAPTER 9. CONCLUSION AND FUTURE WORK 75
9.1 Conclusion 75
9.2 Advantages 75
9.3 Limitations 75
9.4 Future Expansion 76
REFERENCES 77

LIST OF TABLES

No.	Title	Page No.
2.1	Summary of existing works	14
3.1	Definition and Acronyms	17

LIST OF FIGURES

No.	Title	Page No.
3.1	Web scrapping	27
4.1	System architecture diagram	34
5.1	DFD Level 0	38
5.2	DFD level 1	39
5.3	DFD level 2	41
5.4	Individual Work Plan	43
5.5	Gantt chart of Work Plan	43
6.1	Beautiful soup code segment	49
6.2	Chartjs code segment	50
6.3	PdfKit code segment	52
6.4	User authentication module	54
6.5	Placement statistics module(1)	55
6.6	Resume builder module(1)	56
6.7	Resume builder module(2)	57
6.8	Off-campus placement module(1)	58
6.9	On-campus placement module(1)	59
6.10	Roadmap and resources module(1)	60
7.1	Home page	63
7.2	Student profile	63
7.3	sign up for student	64
7.4	login for student	64
7.5	sign up for placement officer	65
7.6	login for placement officer	65
7.7	On campus job home page	66
7.8	On campus job details page	66
7.9	placement officer input page	67
7.10	Applied strudent details viewing page	67
7.11	Off campus job home page	68
7.12	Placement statistics viewing page	69
7.13	Resume input form(1)	69
7.14	Generated resume	70
7.15	Roadmap and resources page(1)	71
7.16	Roadmap and resources page(2)	71

CHAPTER 1
INTRODUCTION

In today's tough job market, students often struggle to find job placements because there's not enough guidance, clear plans, or sufficient opportunities. Currently, the process of landing a job is quite disorganized. Students have to search for jobs, figure out their career path, and find resources separately. Moreover, the manual nature of many placement tasks contributes to delays and errors in communication, interview scheduling, and result tracking. This not only hampers the overall effectiveness of the placement process but also creates a potential for missed opportunities and miscommunication among stakeholders. Additionally, the absence of a structured analytics and reporting system makes it challenging for administrators to extract meaningful insights from placement data, hindering informed decision-making for continuous process improvement.

To solve these issues, we're planning to create a user-friendly website. This website will bring together guidance, career plans, and resources in one place. The goal is to make it easier for students to find and pursue job opportunities. The website aims to simplify the job-hunting process and give students the support they need to navigate the job market successfully.

1.1 PROBLEM DESCRIPTION

The existing procedures for managing placements are outdated and rely heavily on manual efforts for collecting, organizing, and managing information. The entire process of announcing placements and collecting data involves repetitive manual tasks, making it

time-consuming and prone to errors. Additionally, crucial features such as placement statistics, insights from previous interview experiences, and guidance are conspicuously absent from the current setup. Furthermore, information about placements is scattered across various platforms, adding complexity and making it challenging for students and placement officers to access consolidated and comprehensive details. The inefficiencies inherent in the traditional approach highlight the need for a more streamlined and technology-driven solution to enhance the overall placement process.

CHAPTER 2
LITERATURE REVIEW

Placement portals are online platforms used by students and job seekers to browse and apply for employment opportunities. These portals are used by companies and organizations to post job openings and connect with potential candidates. There are lot of online platforms available today such as Naukri, linkedin etc. Each one of them have different Advantages and Disadvantages.

2.1 Naukri[1]

DOMAIN

Naukri is a prominent online job portal in India that connects job seekers with employers by providing a platform for job postings, resume submissions, and career related services.

FUNCTIONALITIES

- Naukri allows job seekers to search for jobs based on various criteria such as job title, location, industry, experience level, and more.
- Job seekers can create detailed profiles with information about their skills, education, work experience, and other relevant details.
- Naukri provides job recommendations to users based on their profiles and preferences, helping them discover relevant job opportunities.
- Naukri provides analytics and insights to help employers understand the effectiveness of their job listings and recruitment efforts.

TECHNIQUES USED BY THE APP

- The website uses machine learning for job recommendations and job alerts.
- The profiles created by the user are stored in the database for further needs in the future.
- Recruiters can use data analytics to evaluate new candidates. For example, by pinpointing the company's top performers and identifying their hard and soft skills
- The paper evaluates the framework and the prototype application using both quantitative and qualitative methods, such as usability tests, questionnaires, interviews, and analytics.

MERITS

- Since the website provide job recommendations and filtering it is easy for job seekers to find their eligible jobs.
- The website helps the recruiters to find the best people from the industry to get recruited in their company.
- A company can make their own brand.
- The website comes with attractive and responsive user interface which is easy to use.

DRAWBACK

- Since there are more number of job seekers it is very competitive to get a job as all of the mentioned jobs are off campus.
- There is no interview experience that can be shared by individual who have been placed.
- The jobs are only available if the recruiter posts them.
- The website is not specific to an institution or university there cannot be placement statistics.

2.2 Linkedin[2]

DOMAIN

LinkedIn is a professional social networking platform that facilitates connections, job opportunities, and business networking for individuals and organizations.

FUNCTIONALITIES

- Linkedin allows people to connect to each other and build professional connections with the best in the industries.
- Provide opportunities for applying of job and internship which are provided by the recruiters and individuals.

- Allows to post about personal achievements and making it allow to see in the public for more recognition.
- Connecting the recruiters and HR through messaging.

TECHNIQUES USED BY THE APP:

- Sockets : Linkedin uses sockets to communicate in the messaging part.
- Machine Learning: Linkedin uses machine learning to personalize your profile and provide recommendations for connecting people and best suite job opportunities as per user need.
- Hootsuite's OwlyWriter AI : This AI technology in linkedin is used to provide attractive captions for social media post.It can also be often used in job application posts to make it more attractive.
- Espresso: powers mission-critical LinkedIn application including member profiles. It is an online, distributed, fault-tolerant NoSQL database.

MERITS

- Helps to build a brand by creating a company profile and share industry updates
- Easier communication through messaging so that people can connect easily.
- Job filter is provided for sorting jobs according to our needs such as location, salary, work experience etc.

- Building professional communication becomes easier.

DRAWBACK

- Since more people are using the application there is always a problem of recognizing as people have to stand out from others.
- Subscription is needed for premium to make the profile stand out from others thus it is costly sometimes.
- No placement statistics as it is not specific to college or university. There are no specific roadmap or guided paths provided for getting placement and job opportunities.

2.3 Geeks for geeks[5]

DOMAIN

Geeks for Geeks is a popular online platform that provides a wide range of computer science and programming resources, tutorials, and articles to help individuals learn and enhance their technical skills.

FUNCTIONALITIES

- The website provides various educational contents tutorials, articles, and guides covering a wide array of computer science, programming, and technology topics.

- Geeks for Geeks offers coding challenges, competitive programming exercises, and a coding platform for practicing and improving coding skills.
- There is also a discussion portal for solution discussions of coding problems.

TECHNIQUES USED BY THE APP

- Search and Navigation: Geeks for Geeks uses advanced search algorithms and filters to help users find relevant content quickly and easily.
- User Accounts and Profiles: User authentication and account management systems are in place to enable features like bookmarking articles, tracking progress, and participating in forums.
- Machine Learning Recommendations: The platform may use machine learning algorithms to recommend relevant articles, courses, or coding problems based on user behavior and preferences.

MERITS

- The website is free and easily accessible to everyone.
- Since there exist a vast community it is easy to solve doubts relating to any topic as people can post and respond to doubts quickly.
- The website also provides interview preparation roadmap and interview experience so that it can be beneficial for those who are preparing for placements and internships.

DRAWBACKS

- The website mainly focuses on steps to actually get a job and the study materials needed to crack a job but provide less job opportunity or openings.
- The content of website is bulky as there are vast resources thus it becomes difficult to get a structured content that are only required to get a job.
- The chances of getting hired is comparatively less as compared to normal placement scenario as competition is very high and job openings are less.

2.4 Glassdoor[6]

DOMAIN

Glassdoor is a popular job search and company review website where employers can create company profiles and post job listings, and job seekers can search for jobs, read company reviews, and access salary information.

FUNCTIONALITIES

- Users can provide reviews about the company where they are working.
- Suggests certain job openings available in the globe.
- Provides resume creating option for user to create their resume.

TECHNIQUES USED BY THE APP

- The application stores the review of users in the database and retrieve it showing to other users.
- The application uses machine learning in job listing by matching it with the skills provided by user or based on user interest.
- The application uses input fields with HTML and CSS for resume creation.

MERITS

- People applying for job need not to create resume from separate website as they are provided in the application itself.
- People can have a better understanding of the company as they can read the review provided from user who have already worked there.
- People can apply for jobs easily as they do not need to have to visit companies directly.

DRAWBACKS

- The application can get only job openings if the recruiter posts a job.
- Does not provide study materials or roadmap that are essential to crack a job.
- Does not show placement statistics of people placed as it is not specific to an institution or university.

2.5 Placement india[7]

DOMAIN

Placement India is a job portal website in India that focuses on job postings, job search, and career-related services for job seekers and employers. These websites typically provide a platform for job seekers to search for job vacancies and for employers to post job listings and recruit suitable candidates. FUNCTIONALITIES

- Filtering of jobs based on location, salary, company, job role and experience etc.
- Job alerts to receive job notifications when listing matches with criteria.
- Provides messaging facility for job seekers to connect to recruiters.
- Provides job posting for recruiters.

TECHNIQUES USED BY THE APP

- Sockets is mainly used for messaging functionality.
- Machine learning to recommend certain job listings according to job seekers skills by matching them.
- Integration with external APIs for features like social media login, job posting syndication or resume parsing.

MERITS

- The website provides wide range of job listings.

- The website provides user friendly interface.
- Since there is job filters it is easy for job seekers to select the required job where they can apply.

DRAWBACKS

- There are large number of users thus competition is comparatively high.
- There are chances of scam and unsolicited messages leading to inaccurate communication.
- There is a subscription cost to enhance profile and getting advanced features.
- There is no separate functionality interview experience or feedback sharing option.

2.6 SUMMARY OF EXISTING WORKS

RELATED WORKS	TECHNIQUES	MERITS	DEMERITS
Linkedin	• Linkedin uses machine learning to personalize your profile and provide recommendations for connecting people and best suite job opportunities as per user need • Hootsuite's Owly-Writer AI : This AI technology in linkedin is used to provide attractive captions for social media post.It can also be often used in job application posts to make it more attractive	• Helps to build a brand by creating a company profile and share industry updates • Easier communication through messaging so that people can connect easily.	• Advanced features require a paid subscription. • No placement statistics as it is not specific to college or university
Glassdoor	• The application stores the review of users in the database and retrieve it showing to other users. • The application uses machine learning in job listing by matching it with the skills provided by user or based on user interest	• People can apply for jobs easily as they do not need to have to visit companies directly. • People applying for job need not to create resume from separate website as they are provided in the application itself.	• The application can get only job openings if the recruiter posts a job. • Does not provide study materials or roadmap that are essential to crack a job
Geeks For Geeks	• The platform may use machine learning algorithms to recommend relevant articles, courses, or coding problems based on user behavior and preferences. • Geeks for Geeks uses advanced search algorithms and filters to help users find relevant content quickly and easily.	• The website also provides interview preparation roadmap and interview experience so that it can be beneficial for those who are preparing for placements and internships. • The website is free and easily accessible to everyone.	• The chances of getting hired is comparatively less as compared to normal placement scenario as competition is very high and job openings are less. • The content of website is bulky as there are vast resources thus it becomes difficult to get a structured content that are only required to get a job.

RELATED WORKS	TECHNIQUES	MERITS	DEMERITS
Placement India	• Machine learning to recommend certain job listings according to job seekers skills by matching them. • Integration with external APIs for features like social media login, job posting syndication or resume parsing.	• The website provides wide range of job listings. • The website provides user friendly interface.	• There are large number of users thus competition is comparatively high. • There is no separate functionality interview experience or feedback sharing option.
Naukri.com	• The profiles created by the user are stored in the database for further needs in the future. • The website uses machine learning for job recommendations and job alerts.	• Since the website provide job recommendations and filtering it is easy for job seekers to find their eligible jobs. • The website helps the recruiters to find the best people from the industry to get recruited in their company.	• Since there are more number of job seekers it is very competitive to get a job as all of the mentioned jobs are off campus. • The jobs are only available if the recruiter posts them.

Table 2.1: Summary of existing works

2.7 MOTIVATION FOR THE PROJECT PROPOSAL

The main motivation for developing the website is to enhance the current placement procedure of campus as there is only basic mail facility for on campus placements. The website is developed by analysing the merits and demerits of existing solution by automating all the conventional placement procedure and integrating the all level of users in a single platform for more convenient and efficient result.

2.8 PROBLEM STATEMENT

The problem statement can be stated as to design and develop a web application which helps in enhancing overall procedure of placement process by including placement progress, data management, on campus and off campus jobs, resume builder, interview experience and placement preparation resources.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 INTRODUCTION

3.1.1 Purpose

A placement portal serves the purpose of connecting students with job and internship opportunities, facilitating the job search process, and helping students get industry exposure. It provides features like job listing, resume building tools, placement preparation resources and a platform for employers to recruit students. It plays a crucial role in helping students transition from their academic studies to meaningful employment opportunities.

3.1.2 Definitions and Acronyms

TERM	DEFINITION
System	• Product and software is used interchangeably to denote the 'Placement Portal GECW' website.
Users	• They include students, alumni and placement officer who use the website.
User's Data	• Data of the Users saved in our database.
Admin	• A person or a group of individuals who control the overall operation of the system.
Rendering HTML	• The process of generating HTML markup to display web pages in the browser.

Table 3.1: Definition and Acronyms

3.2 OVERALL DESCRIPTION

We are building a placement assistant portal as a website exclusively for the students of GECW. This project combines various features of other websites and job portals to identify their pros and create an efficient way for placement assistance both off-campus and on- campus. The necessity of this project is it makes finding jobs easier and faster for students and placement officers. It lets you check placement information anytime, encourages clear communication, and helps colleges handle student information well. The website is adjusted to fit the college's specific needs, it can also get more alumni involved, follow the rules, and grow as needed. The portal also provides a roadmap to placement guiding the student from the start of the college itself.

3.2.1 User Needs

Students: Students are the primary beneficiaries of the website. They will use it to:

- Search and apply for job openings posted by placement officer as well as off campus.
- Create their resumes and academic records.
- Receive notifications about upcoming placement events and deadlines.
- Access resources like roadmap and interview preparation materials.
- View and analyse the placement statistics of college.

Placement Officer:Placement officers play a crucial role in managing the placement process. He/She will use the website to:

- Post job openings on behalf of companies.
- Review and approve student applications.
- Analyse placement data and trends.

Alumni: Alumni can login to the website and share their interview experience they have experienced during their interview.

3.2.2 Assumptions and Dependencies

- Platform Assumption: This website will be available for Android, iOS, Windows, and Mac. This will ensure cross-platform compatibility.

- Browser Assumption: This website will be available in every browser in a responsive manner.

- Internet Connectivity Assumptions: The users will be able to access all the functions of the portal through an internet connection.

- User Skill Assumptions: Users (students, faculty, and placement officers) possess basic computer literacy and internet navigation skills.

- Security Assumptions: Security measures, such as encryption, user authentication, and access controls, will be implemented to protect user data and the website itself.

- Data Availability Assumptions: On-campus data will be available from placement officer and off-campus data will be available from web scraping.

- Programming Language Assumptions: Python is the programming language used in executing the functionalities in the website using django framework where HTML, CSS and bootstrap and JavaScript can be used in front end. Whereas SQLite can be used as the database.

3.3 METHOD OF REQUIREMENT ELICITATION

3.3.1 Brainstorming

All four of us sat down and talked about the project idea. We tried thinking from the point of view of students, placement officer, and alumni. At first, different problems that are faced by students for achieving the placements as there are no proper resources and availability of opportunity as all of these features are scattered in different places . So

we discussed many ideas to make students more comfortable in proceeding with their placement journey. Each group member came up with their points and all of them were noted down. The points the majority thought as relevant were specially marked. Ranks were given to each of the requirements. An elimination round was finally conducted to remove the least relevant requirements and keep the rest.

3.3.2 System Study

We have conducted discussions among colleagues and faculty about the requirements of such a system. We have also surveyed the existing similar applications that are mentioned in the above chapter. We thoroughly understood the current system's features as well as shortcomings.

3.4 SYSTEM FEATURES AND REQUIREMENTS

3.4.1 Functional Requirements

(i) Registration of students, placement officer and alumni

- **Use case name**: Registration of students, placement officers and alumni in the website.
- **Objective** : To include the users in the website and check for verification in later stages.
- **Precondition** : For students it is necessary to have their college register number,

email name, phone number. For placement officer and alumni it is necessary to have name, and email.

- **Trigger**: The user will click on the sign up button for registering in the website.
- **Post condition**: Users who are registered in website will have a unique id assigned to them.
- **Flow of events** : Basic Flow

  Step 1: User(placement officer, students, alumni) open the website.

  Step 2: User register by providing required information or details as mentioned in the registration form.

(ii) Login of existing users

- **Use Case name** : Entry of user into website after registration.
- **Objective** : To authenticate a user by logging in.
- **Precondition**: User should be registered by providing all the required details.
- **Trigger**: User clicks on the login button on getting logged in to the website.
- **Postcondition** : User can access their account.
- **Flow of Events** : Basic Flow

  Step 1: After login, user gets redirected to home page of the website.

  Step 2: User selects required options as per his/her wish for further information.

(iii) Build resume :

- **Use case** : Building resume for applying for both on campus and off campus placements.
- **Objective** : To make a job ready resume for placements which are happening on campus as well as off campus.
- **Precondition** : The user should be a student and he/she must be registered in the website .
- **Trigger**: The user clicks on the build button for creating the resume.
- **Postcondition** : Students can get access to resume builder.
- **Flow of Events** : Basic Flow

  Step 1: Students can select the resume builder option from home page.

  Step 2: Students can fill the required details as mentioned in the form.

  Step 3: Students can download the resume after clicking on build button.

(iv) Manage on campus placements :

- **Use case** : Job openings that are provided by placement officer in the campus.
- **Objective** : To provide on campus job opportunity to students in the campus.
- **Precondition** : The user should be a student and or placement officer and he/she must be registered in the website .
- **Trigger**: The students click on apply button to apply for on campus placements provided by placement officer.

- **Postcondition** : Students can get access to job notification and apply it through the website.

- **Flow of Events** : Basic Flow

  Step 1: Students get the job roles available in the home page.

  Step 2: Students after clicking on apply get redirected to applying phase for providing required details.

(v) Provide Off Campus placements:

- **Use case**: Job openings that are provided apart from campus.

- **O**bjective: To provide on campus job opportunity to students outside of the campus.

- **Precondition**: The student must be registered in the website.

- **Trigger**: The students click on apply button to apply for off campus placements.

- **Postcondition**: Students can get access to job notification and apply it through the website.

- **Flow of Events**: Basic Flow

  Step 1: Students get the job roles available in the home page.

  Step 2: Students after clicking on apply get redirected to company's page for applying.

(vi) Provide roadmap and resources:

- **Use case**: Guided path provided for cracking off campus, on campus placements and internships.
- **Objective**: To give a guided path to students by providing only required resources to crack a placements and internships.
- **Precondition**: The student must be registered in the website.
- **Trigger**: The user clicks on the roadmap and resource option in navigation bar to get access.
- **Postcondition**: Students can get access to roadmap as well as resources available in the website.
- **Flow of Event**: Basic Flow

  Step 1: Students after registration gets to the home page.

  Step 2: Students after clicking on roadmap and resources gets directed to that page.

  Step 3: Students can access the roadmap and free resources as per their need.

(vii) Share alumni interview experience :

- **Use case** : Interview experience of alumni and current seniors who have already placed.
- **Objective** : To give the students the interview exposure by sharing of interview experience of alumni or seniors.
- **Precondition**: The student must be registered in the website.

- **Trigger**: Alumni clicks on share button after entering interview experience.
- **Postcondition**: Students can get access to interview experience
- **Flow of Events**: Basic Flow

  Step 1: Students after registration gets to the home page.

  Step 2: Students after clicking on interview experience get directed to the particular page.

  Step 3: Students can read the interview experience.

(viii) Analyse placement statistics :

- **Use case** : Showcasing of placement statistics of college.
- **Objective** : To give the students idea about current and past placement statistics of the college.
- **Precondition** : The student must be registered in the website .
- **Trigger**: The user clicks on show button to showcase available statistics of placement.
- **Postcondition** : Students can view or get details about the placement statistics about the college.
- **Flow of Events** : Basic Flow

  Step 1: Students get the placement statistics option available in the home page.

  Step 2: Students after clicking on it gets directed to placement statistics .

3.4.2 External Interface Requirements

User Interfaces:

- The application should have an intuitive and user-friendly interface for laptops, computers, desktop and mobile phones
- The user interface should support various screen sizes and resolutions to ensure compatibility with a wide range of devices.

Software Interfaces:

- The website should be compatible with the latest versions of web browsers.

Database Interfaces:

- The website should interact with a robust and scalable database system to store placement information, user details, and job data.
- Ensure efficient data retrieval and storage.

Web Scrapping Services:

- The Website should be integrated with BeutifulSoup library as to use web scrapping services.
- Beautiful Soup provides methods to parse HTML documents, creating a parse tree that represents the structure of the document. This makes it easy to navigate and

manipulate the document's elements and data. In the project we are retrieving data from Placement India website using web scrapping.

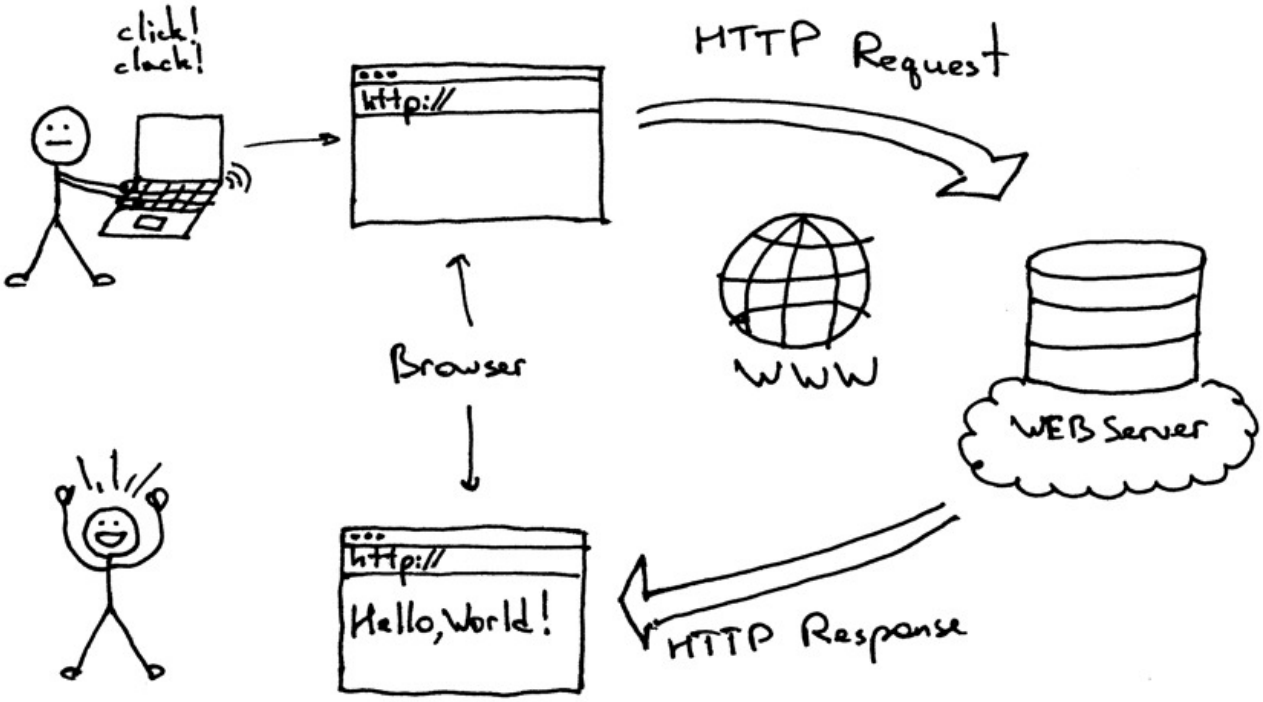


Figure 3.1: Web scrapping

3.4.3 Nonfunctional Requirements

- **Security** : The website uses csrf tokens in django for preventing malicious attacks hence ensures that any requests coming back in are cross-checked against this token. The website also provides authentication of existing users with security via this method.

- **Compatibility** : The website is accessible at any phones irrespective of operating system and their versions. But still the system demands 2 GB of ram in it to have smooth working and also a good internet connectivity.

- **Maintainability and Manageability** : When coming to the maintenance and management part it is required in case of adding any new feature in the website. Maintenance also place a major role if there is any issue in the backend which is fixed later on.

- **Scalability Requirements**: The website is highly scalable as it supports multiple user for registering in the website. Also it provides fast and easy access.

- **Capacity** : The website stores all the information regarding to students, alumni, placement officer in a database. Thus it provide a well structured storage capacity in present and future.

- **Reliability and Availability** : The website is highly reliable as it stores all information in database which can be further verified and can be used to provide information such as placement statistics. The website is available to all who have registered in the website.

3.4.4 Hardware Requirements

Development Machines: Standard laptops used by the development team.It should have a minimum specification of an Intel Core i3 or an equivalent AMD processor and 2GB of RAM.

3.4.5 Software Requirements

Django framework: Used for building robust and scalable web applications using the Python programming language.

SQLite: Integrated for backend services, including authentication and storing data in the database.

Python: Utilized as the primary language for developing website.

HTML, CSS and JavaScript: For developing the front end of website.

Visual Studio Code: Chosen as the primary integrated development environment (IDE) for website development.

Git and GitHub: Used for version control and collaboration among development team members.

BeautifulSoup: Python library used for web scrapping.

ChartJs: JavaScript libray for making HTML-based charts for data visualization.

3.5 PROCESS MODEL

Scrum Agile Development Model:

Iterative Development:

Scrum breaks down the development process into short iterations called sprints, usually lasting 2-4 weeks. This allows your team to release a functional version of the application at the end of each sprint, providing continuous value to users. Adaptability to Changes:

Agile methodologies, including Scrum, are designed to accommodate changes in requirements even late in the development process. This is crucial for a project like yours, where

user feedback and evolving market trends may prompt adjustments to features or priorities.

User-Centric Approach:

Agile emphasizes collaboration with stakeholders, including end-users. Regular sprint reviews and feedback sessions provide opportunities for users to actively participate in shaping the application, ensuring that it meets their needs and expectations.

Cross-Functional Teams:

Scrum promotes the formation of cross-functional teams with members possessing diverse skills. In the context of your travel app, this could include developers, UX/UI designers, language specialists, and travel experts working collaboratively.

Continuous Integration and Testing:

The Agile model encourages continuous integration of code and regular testing throughout the development process. This ensures that the application remains stable and functional as new features are added.

Prioritization of Features:

Agile methodologies use product backlogs to prioritize features based on their importance and value. This allows your team to focus on implementing the most crucial features first, providing a Minimum Viable Product (MVP) early in the development cycle.

Increased Transparency:

Regular sprint planning, daily stand-up meetings, and sprint reviews foster transparency within the development team. This transparency helps manage expectations and keeps all stakeholders informed about the project's progress.

Quick Time-to-Market:

The iterative and incremental nature of Scrum allows for a quicker time-to-market. This can be beneficial in the competitive landscape of mobile applications, enabling you to release valuable features and updates rapidly.

3.6 FEASIBILITY STUDY

3.6.1 Economic Feasibility

The economic analysis affirms the cost-effectiveness and justification of the GECW Placement Portal. Its numerous advantages, including streamlined placement processes, elevated student success rates, and efficient recruitment for employers, significantly outweigh the financial investments required.

3.6.2 Technical Feasibility

The project team possesses the essential technical resources and expertise required to conceptualize and deploy the GECW Placement Portal. The necessary hardware and software components are not only feasible but also adaptable, ensuring seamless scalability for future enhancements.

3.6.3 Operational Feasibility

Aligned seamlessly with the operational necessities of the college, the GECW Placement Portal will augment daily activities by simplifying intricate placement-related processes. Minor modifications to existing workflows will ensure the portal's seamless integration into the college's operational ecosystem.

3.6.4 Scheduling Feasibility

The project schedule has been meticulously planned, ensuring a well- defined timeline for each development phase. Clearly delineated milestones and deadlines are in place, guaranteeing efficient project management and timely completion.

CHAPTER 4
ARCHITECTURAL DESIGN

Architectural design focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process. The diagrammatic representation of the system architecture is called the system architecture diagram. This diagram gives us the abstract view of the components and their relationship with the system that makes the system work.

The architecture diagram of the system consist of authentication, resume builder module, offcampus and oncampus module, interview experience module, placement statistics module and roadmap and resources module. The components related to each module or the functional requirements is given in the below system architecture diagram.

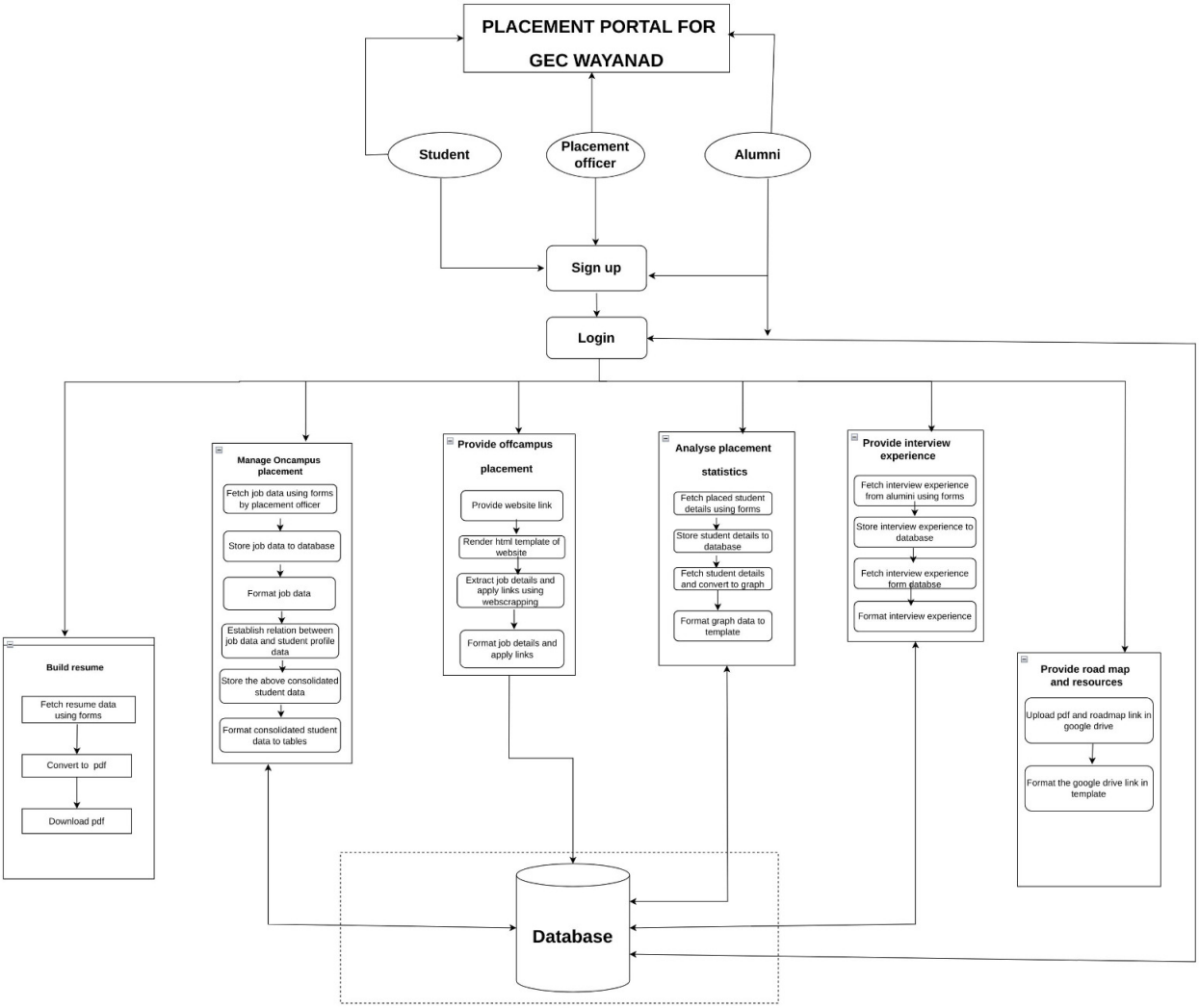


Figure 4.1: System architecture diagram

4.1 MODULE DESCRIPTION

Module name	**Input data**	**Action/Process**	**Output data**
Authentication module	Users credentials.	-Registration and authentication of user is done.	Dashboard.
On Campus placement module	Job details and apply links,Student application data	-Collect job details and description using form by placement officer. -Store job data in database -Format job data -Collect student application details from profile and establish relation between above job data and store in database	-Formatted job details and description. -Consolidated student data.
Off campus placement module	Off campus website link.	-Render html template for the website Placement India. -Extract job details and apply links using web scrapping.	Formatted job details and apply links.

Module name	**Input data**	**Action/Process**	**Output data**
Placement statistics module	Placed student details	-Collect placed students details using forms. -Store student details to database. -Fetch student details and sort according to fields	Convert data to graph and display in template.
Resume builder module	Personalised student data for resume.	-Collect resume data using forms. -Transfer to a standard html template. -Convert to pdf format for downloading	Standardized resume in pdf format.
Interview experience module	-Alumni interview experience.	-Collect interview experience data by forms. -Store interview experience to database. -Fetch interview experience from database and format.	-Formatted interview experience.
Roadmap and resources module	-Usefull study materials required for placements.	-Materials are stored in google drive storage with public access.	-A single page consisting of all the materials for the user to view or download.

CHAPTER 5
DESIGN AND IMPLEMENTATION

5.1 INTRODUCTION

Here we are giving the detailed design of the proposed system using data flow diagrams. Here we are discussing about the level 0, level 1, level 2 data flow diagram of the system to explain the processess and data flow involved in the system.

A Data Flow Diagram (DFD) is a visual representation of how data flows within a system. It is a graphical tool used in software engineering to illustrate the flow of data between various processes, data stores, and external entities. DFDs are a part of structured analysis and design methods and are particularly useful during the early stages of software development.

5.2 DATA FLOW DIAGRAMS

A data flow diagram (DFD) is a graphical representation of the flow of data within an organization or system. It is used to show how data moves between processes, stores, and external entities. DFDs are commonly used in system analysis and design to visualize and understand the information flow within a system. They are also helpful for identifying potential bottlenecks and inefficiencies in data processing.

5.2.1 DFD level 0

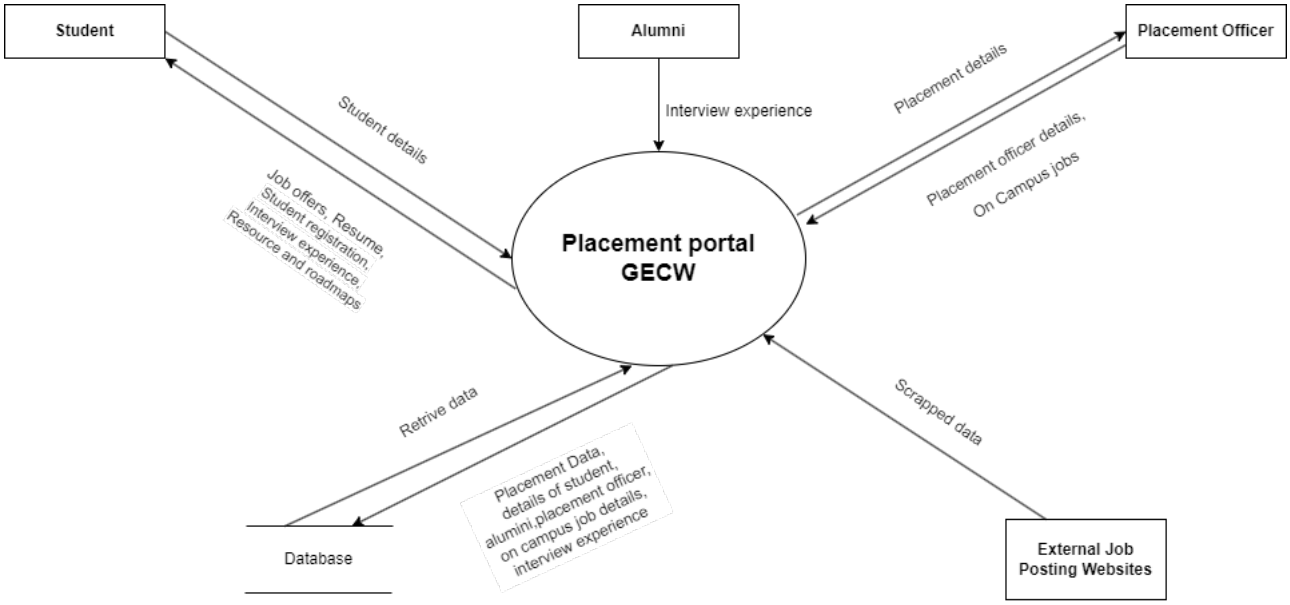


Figure 5.1: DFD Level 0

This level 0 data flow diagram explains Placement Portal for GECW as the central system interacting with external entities. The entities in the system are students, placement officer, alumni. All the users get logged into the website and can access the features. Placement officer can post on campus job details, and can get applied student details. Alumni can share interview experience from the website. Student are the major beneficiary of the website where they can get on campus and off campus job details and apply for them. Students can also access features such as placement statistics, roadmap and resources, resume builder, interview experience by alumni.

5.2.2 DFD level 1

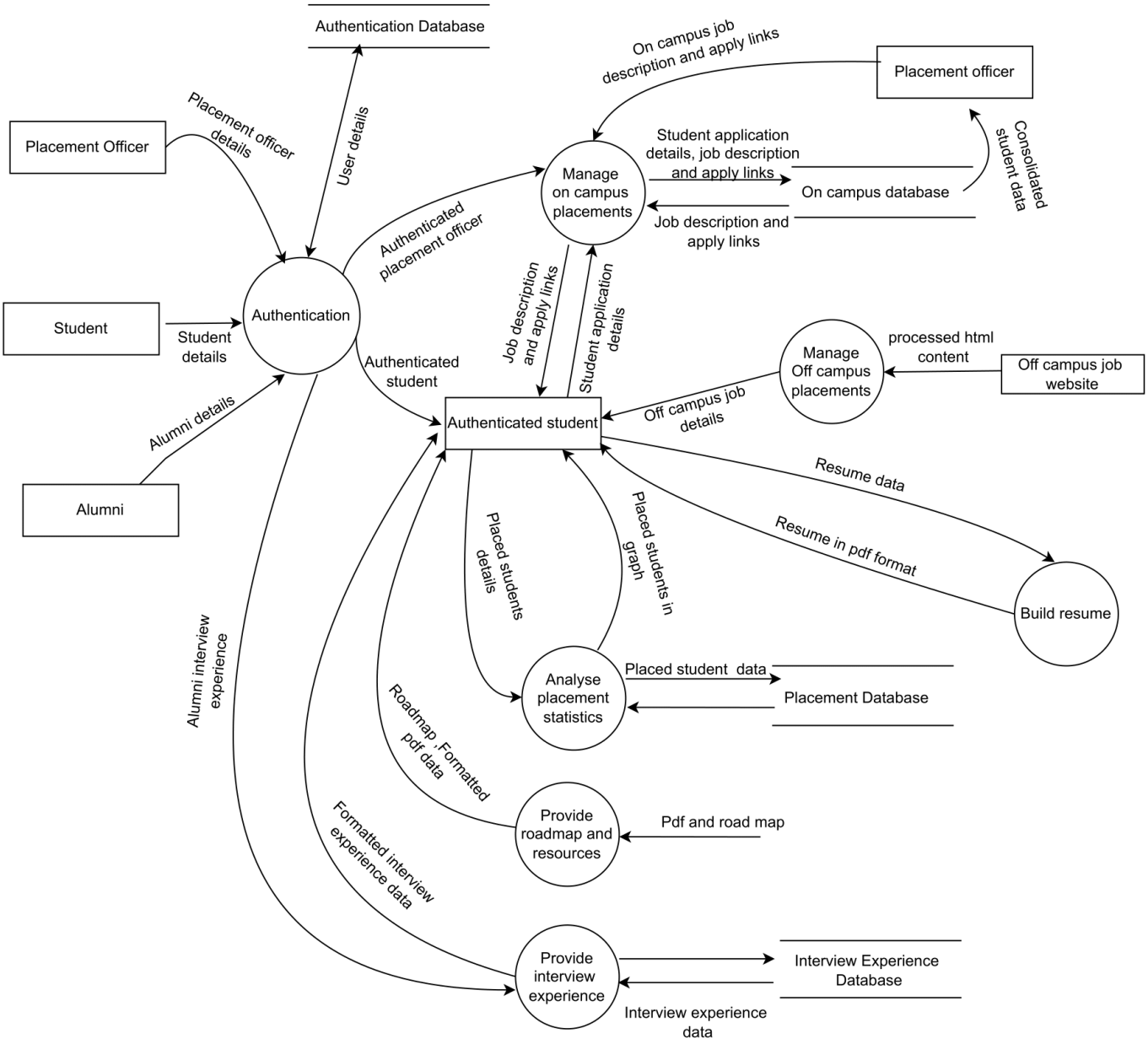


Figure 5.2: DFD level 1

This level 1 data flow diagram explains in detail about all the module available in the website. The first module is the authentication which authenticate all the users who use the website. The resume builder module is used to build resume for students for applying for on campus and off campus jobs. The placement statistics shows the current placement statistics of college in graphical format. The resource and roadmap provides resources for cracking a job and provide a guided path to crack them. The interview experience module gives the student interview experience by alumni. The on campus module provide on campus job opportunity provided by placement officer to students where students can apply to on campus jobs. The off campus module provides off campus jobs that are applicable for students.

5.2.3 DFD level 2

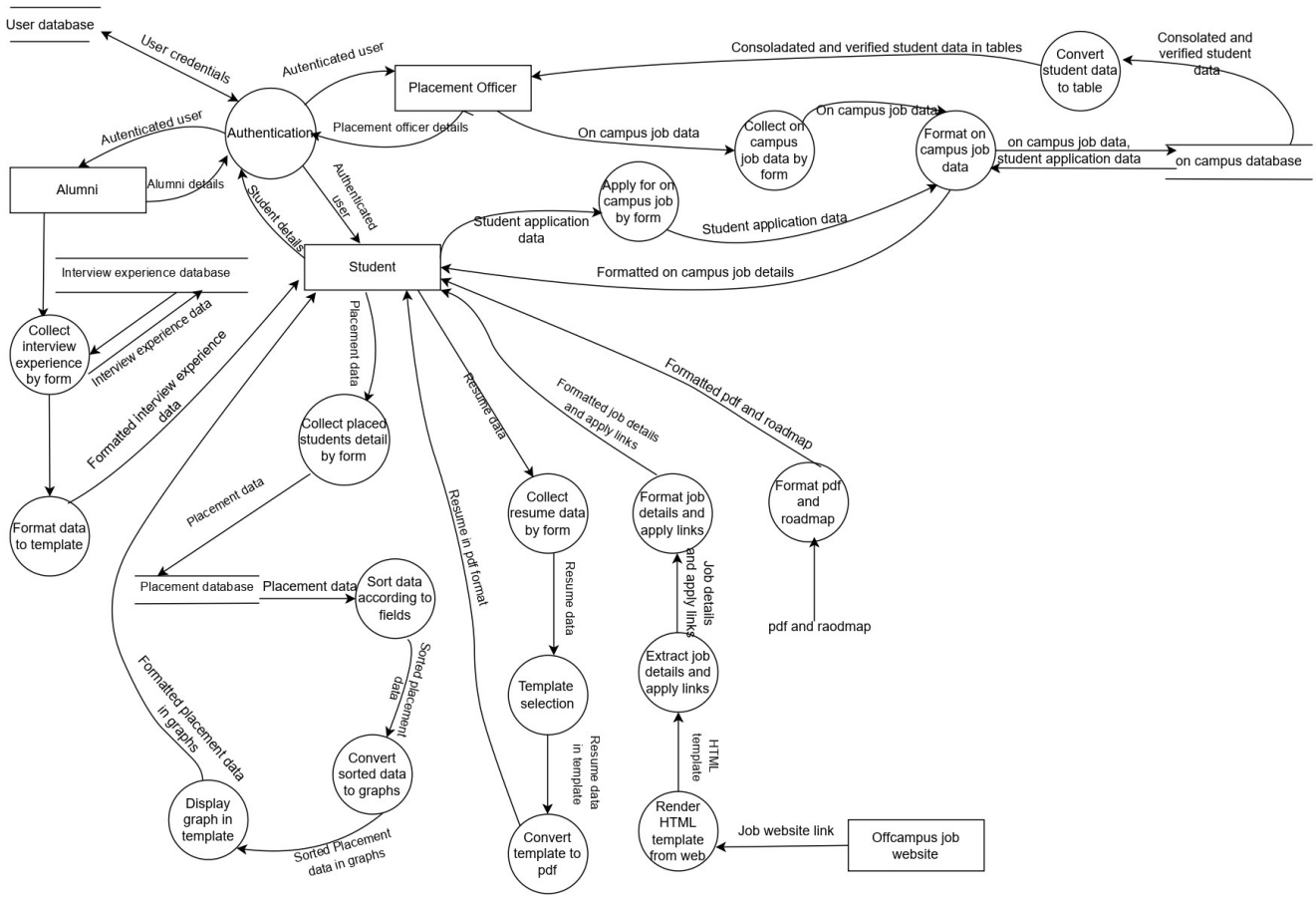


Figure 5.3: DFD level 2

This level 2 data flow diagram explains in detail about each module in the website. The authentication module is divided into 2 parts where users first sign up by creating an account. Then the user logs into website to use the available functionality in the website. The user can only use the functionalities after logging in.

The interview experience module first accepts data from alumni,stores them in database and makes the data in better format and display data to students. The on campus module first accepts job data from placement officer by form, stores them in database and format them in better way and displayed to students. The placement officer also gets list of applied students from it.

The placement module accepts placed students data as input and stores them in database the data is then converted to graph and formatted to template and displayed to students. The resume builder module first accepts resume data from students by form, then the data is copied to an HTML template and it is converted to pdf format for downloading.

The off campus placement module accepts job website link then the HTML template of provided website is rendered and job details with apply links are extracted. The extracted details and links are formatted and displayed to students. The roadmap and resource contain static data of pdf data and roadmap which is formatted and displayed to students.

5.3 WORK PLAN AND TASK ALLOCATION

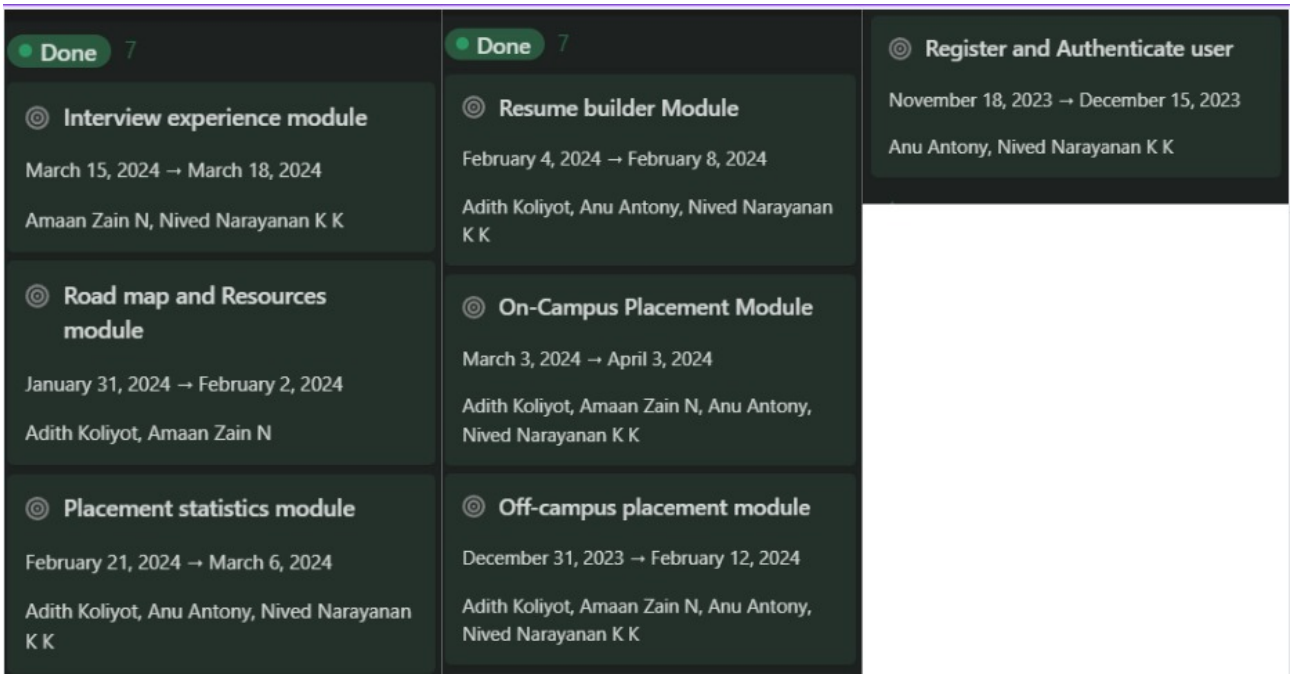


Figure 5.4: Individual Work Plan

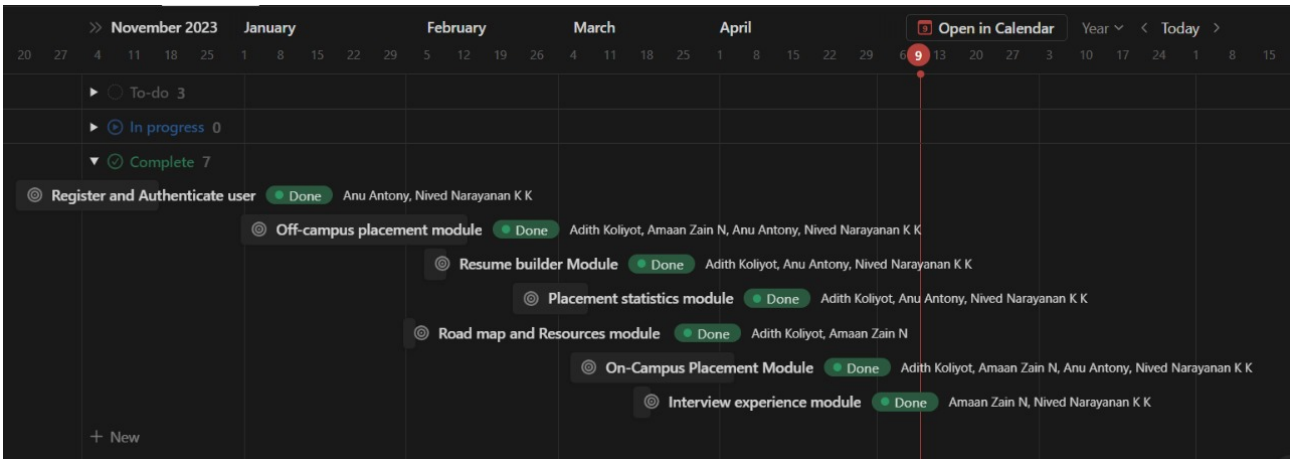


Figure 5.5: Gantt chart of Work Plan

CHAPTER 6
CODING

6.1 INTRODUCTION

Our project is a web application tailored for Government Engineering College in Wayanad, focusing on creating a placement portal for students and recruiters. To make this happen, we're using HTML, CSS, JavaScript, Bootstrap, Django, and SQLite to build a strong and reliable platform.

At the core of our application is Django, a Python framework known for its scalability. It helps us manage student registration, profiles, job postings, and interactions with recruiters. To ensure that our data is stored securely and efficiently, we're using SQLite as our backend database.

Our portal will have a user-friendly interface, developed using HTML, CSS, and Bootstrap. Students will be able to easily set up profiles, upload resumes, and explore job opportunities. Recruiters will have the ability to post jobs, review student profiles, and schedule interviews.

We're also incorporating JavaScript to add interactivity to our portal, allowing for features like real-time notifications and form validation. Plus, we're making sure that our application is compatible with various devices by implementing responsive design principles.

In our project, we're utilizing Beautiful Soup for web scraping tasks. Beautiful Soup is a Python library that enables us to extract data from HTML and XML files by providing

convenient methods for navigating and searching the parse tree.

Ultimately, our goal is to create a placement portal that meets the specific needs of Government Engineering College, Wayanad, while also setting a new standard for usability and efficiency in the realm of placement portals.

6.1.1 Frameworks

(i) Django

Django is a high-level web framework written in Python, designed to facilitate rapid development and maintainability of web applications. It promotes the "Don't Repeat Yourself" (DRY) principle, allowing developers to build applications efficiently by minimizing redundant code. At its core, Django provides a comprehensive set of components for common web development tasks, including URL routing, database management, form handling, and authentication. Its Object-Relational Mapping (ORM) layer simplifies database interaction by abstracting away the complexities of SQL queries, enabling developers to define database models using Python classes.

Furthermore, Django offers a powerful templating engine that facilitates the creation of dynamic web pages by embedding Python code within HTML templates. This seamless integration of backend data with the presentation layer of the application enables developers to build rich and interactive user interfaces. With built-in security features such as cross-site scripting (XSS) protection, cross-site request forgery (CSRF) protection, and authentication mechanisms, Django ensures that applications are secure by default. Its adherence to the Model-View-Template (MVT) architectural pattern

streamlines development and promotes code organization, making Django a popular choice for building a wide range of web applications.

6.1.2 Tools

(i) Visual Studio Code

Visual Studio Code (VS Code) is a free source-code editor developed by Microsoft for Windows, Linux, and macOS. It provides developers with a lightweight but powerful environment for writing, debugging, and running code across various programming languages.

VS Code offers a wide range of features, including syntax highlighting, code completion, linting, debugging, and version control integration (such as Git). It also supports extensions that allow users to customize and enhance their coding experience with additional features and functionalities.

One of the key strengths of VS Code is its flexibility and extensibility. Users can install extensions for different programming languages, frameworks, and tools, enabling them to tailor the editor to their specific needs and preferences. Additionally, VS Code is known for its fast performance and responsiveness, making it a popular choice among developers for a wide range of projects, from small scripts to large-scale applications.

(ii) Git and Git Hub

Git is an open sourced version control tool widely used among developers. It allows the developers to easily manage code versions. It provides functionalities to store every

version of the code and switch between any version as necessary. Git is used as the version control tool for this project.

Git Hub adds cloud functionalities to Git. This enables multiple individuals to work on a single code base simultaneously. The changes made by each user is tracked and synced between every user who works on the project. Git Hub acts as a cloud repository for the project.

6.1.3 Library

(i) Beautiful Soup

Beautiful Soup is a Python library designed for web scraping tasks. It simplifies the process of extracting data from HTML and XML documents by providing convenient methods for navigating the parse tree and searching for specific elements. With Beautiful Soup, developers can easily parse and extract information from web pages, making it a valuable tool for tasks such as data mining, content aggregation, and automated testing.

The library parses the markup of a webpage and provides a data structure representing the HTML/XML document, allowing developers to access and manipulate its contents using Python code. Beautiful Soup's intuitive API and powerful features, such as the ability to handle invalid markup and navigate complex document structures, make it popular among developers for a wide range of web scraping projects.

Overall, Beautiful Soup simplifies the process of extracting data from web pages, enabling developers to quickly and efficiently gather information for analysis or integration into other applications. Its flexibility, ease of use, and robust functionality

make it a valuable tool for web scraping tasks in Python projects.

In our off-campus placement module, we're employing web scraping techniques to gather information about job opportunities from external websites. By utilizing tools like Beautiful Soup, we can extract relevant data from these sites, such as job titles, company names, job descriptions, and application deadlines. Once extracted, we standardize this information into a consistent format within our project, ensuring uniformity and ease of access for users.

Through web scraping, we automate the process of collecting off-campus job listings, saving users valuable time and effort in searching for opportunities. The standardized format we apply to the extracted data streamlines the presentation and management of job listings within our platform. This standardized format might include structured fields like job title, company, location, and a direct link to the job posting.

By integrating web scraping into our off-campus placement module, we provide users with a comprehensive and up-to-date database of job opportunities. This enables them to stay informed about available positions and facilitates their engagement with the off-campus job market, ultimately enhancing their chances of securing relevant employment opportunities.

```
page=requests.get('https://www.placementindia.com/job-search/search.ph
soup=BeautifulSoup(page.text,'html.parser')
role_list=[]
company_list=[]
exp_list=[]
ctc_list=[]
l_list=[]
jd=JobDetails.objects.all()
if(jd!=None):
    jd.delete()
for link in soup.find_all('a',class_='job-name'):
    role=link.text
    role_list.append(role)
```

Figure 6.1: Beautiful soup code segment

(ii) Chart.js

Chart.js is a JavaScript library that empowers developers to create visually stunning and interactive charts and graphs for web applications. Leveraging the power of HTML5 canvas, it offers a simple yet robust solution for generating various types of charts, such as line, bar, pie, and more. With its intuitive API and extensive documentation, developers can easily customize the appearance and behavior of charts to suit their specific needs. Chart.js excels in responsiveness and interactivity, enabling seamless integration of dynamic data visualization into web projects. It has become a popular choice among developers for its ease of use, flexibility, and ability to deliver compelling data-driven experiences to users.

We're integrating Chart.js into our project to visually represent the placement statistics of previous years. This feature allows us to showcase essential metrics such as the number of placements, average CTC (Cost to Company), and distribution across

different branches or departments. Furthermore, Chart.js enables us to implement sorting functionalities, empowering users to organize the data according to various criteria such as branch, department, or CTC range. By leveraging the capabilities of Chart.js, we aim to provide users with an intuitive and informative visualization of placement data, facilitating easy analysis and comparison across different parameters.

```
const ctx1= document.getElementById('dept_chart');
new Chart(ctx1, {
  type: 'doughnut',
  data: {
    labels: {{dept|safe }},
    datasets: [{
      label: 'No of students placed',
      data:{{dept_count|safe}},
      borderWidth: 1
    }]
  },
  options: {

  }
});
```

Figure 6.2: Chartjs code segment

(iii) PDFKit

PDFKit is a JavaScript library that enables developers to generate PDF documents dynamically on the server-side using Node.js. It provides a straightforward and flexible API for creating and manipulating PDF files, allowing users to add text, images, shapes,

and other elements to the document. PDFKit simplifies the process of generating PDFs programmatically, making it easy to customize the content and layout according to specific requirements. Additionally, it supports features such as embedding fonts, adding metadata, and creating bookmarks, ensuring that the generated PDFs meet the desired standards and specifications. Overall, PDFKit is a powerful tool for automating the creation of PDF documents in web applications and server-side environments.

In our project's resume builder module, we're utilizing PDFKit to implement a standardized resume template. This template serves as a foundational structure for generating professional resumes, ensuring consistency and coherence across all documents. With PDFKit, we can dynamically populate this template with user-provided information, such as personal details, education history, work experience, and skills.

By leveraging PDFKit for our standardized resume template, we streamline the resume creation process for users, eliminating the need for manual formatting and layout adjustments. This results in efficient and visually appealing resumes that effectively showcase individuals' qualifications and experiences to potential employers.

```
template=loader.get_template('Student/trial.html')
html=template.render({'profile':profile,'sk_list':sk_list})
options={
    'page-size':'Letter',
    'encoding':"UTF-8",
    'enable-local-file-access': "",
}
pdf=pdfkit.from_string(html,False,options)
response=HttpResponse(pdf,content_type='application/pdf')
response['Content-Disposition']='attachment'
filename="resume.pdf"
```

Figure 6.3: PdfKit code segment

(iv) Requests

The ‘requests‘ library in Python is a popular HTTP library used for making HTTP requests in web applications. It simplifies the process of sending HTTP requests and handling responses, allowing developers to interact with web services and APIs easily.

With the ‘requests‘ library, developers can perform various HTTP operations such as GET, POST, PUT, DELETE, and more. It provides a simple and intuitive API for constructing requests, setting headers, passing parameters, and handling cookies.

One of the key features of the ‘requests‘ library is its ability to handle sessions, allowing developers to persist cookies across multiple requests within the same session. This is particularly useful for scenarios where authentication is required or when interacting with web services that maintain state.

Additionally, the ‘requests‘ library supports features like SSL verification, timeouts,

streaming responses, and automatic decompression, making it suitable for a wide range of use cases in web development, data scraping, automation, and more.

Overall, the ‘requests‘ library simplifies the process of working with HTTP in Python, providing developers with a powerful and versatile tool for making HTTP requests and interacting with web services effectively.

In our project, we’re leveraging the ‘requests‘ library to handle HTTP requests efficiently. By integrating ‘requests‘ into our codebase, we can easily communicate with external APIs, web services, or other HTTP endpoints to fetch or send data as needed.

Using ‘requests‘, we’re able to construct various types of HTTP requests such as GET, POST, PUT, DELETE, etc., and include headers, parameters, and payloads as required by the API endpoints we’re interacting with. This allows us to retrieve information, submit data, or perform actions seamlessly within our application.

Additionally, ‘requests‘ provides features like session management, authentication handling, and automatic response parsing, which streamline the process of working with HTTP in our project. These capabilities ensure that our HTTP interactions are reliable, secure, and efficient.

By incorporating the ‘requests‘ library into our project, we’re able to maintain clean and concise code while effectively managing HTTP communication, ultimately enhancing the functionality and usability of our application.

6.2 SAMPLE CODES

```
def s_login(request):
    message=""
    if request.method=='POST':
     username=request.POST['username']
     password=request.POST['password']
     user=auth.authenticate(username=username,password=password)
     if user is not None:
       auth.login(request,user)
       print("user verified",user.username)
       return redirect('/StudentHome')
     else:
       message="Invalid credentials"
    return render(request,'Student/login.html',{'message':message})

def s_logout(request):
  auth.logout(request)
  print("logged out")
  return redirect('/StudentLogin')

def s_signup(request):
   if(request.method=="POST"):
    first_name=request.POST['firstName']
    last_name=request.POST['lastName']
    username=request.POST['username']
    password1=request.POST['password1']
    password2=request.POST['password2']
    email=request.POST['email']
    department=request.POST['department']
    batch=request.POST['batch']
    user=User.objects.create_user(username=username,password=password1,first_name=first_name,last_name=last_name,email=email)
    profile=Profile.objects.create(username=username,user=user,name=first_name+" "+last_name,email=email,batch=batch,department=department)
    print("User saved")
    return redirect('/StudentLogin')
   else:
    return render(request,'Student/signup.html')
```

Figure 6.4: User authentication module

```
def splacement_stats (request):
    result1= PlacementDetails.objects.values('department').annotate(total=Count('department'))
    result2= PlacementDetails.objects.values('batch').annotate(total=Count('batch'))
    result3= PlacementDetails.objects.values('company_name').annotate(total=Count('company_name'))
    dept = []
    dept_count= []
    for item in result1:
      dept.append(item['department'])
      dept_count.append(item['total'])
    dept_data=zip(dept,dept_count)
    company=[]
    company_count=[]
    for item in result3:
       company.append(item['company_name'])
       company_count.append(item['total'])
    company_data=zip(company,company_count)
    batch=[]
    batch_count=[]
    for item in result2:
       batch.append(item['batch'])
       batch_count.append(item['total'])
    batch_data=zip(batch,batch_count)
    return render(request,'Student/placementstats.html',{'dept':dept,'dept_count':dept_count,'company':company,
                  'company_count':company_count,'batch':batch,'batch_count':batch_count,'dept_data':dept_data,
                  'company_data':company_data,'batch_data':batch_data})
```

Figure 6.5: Placement statistics module(1)

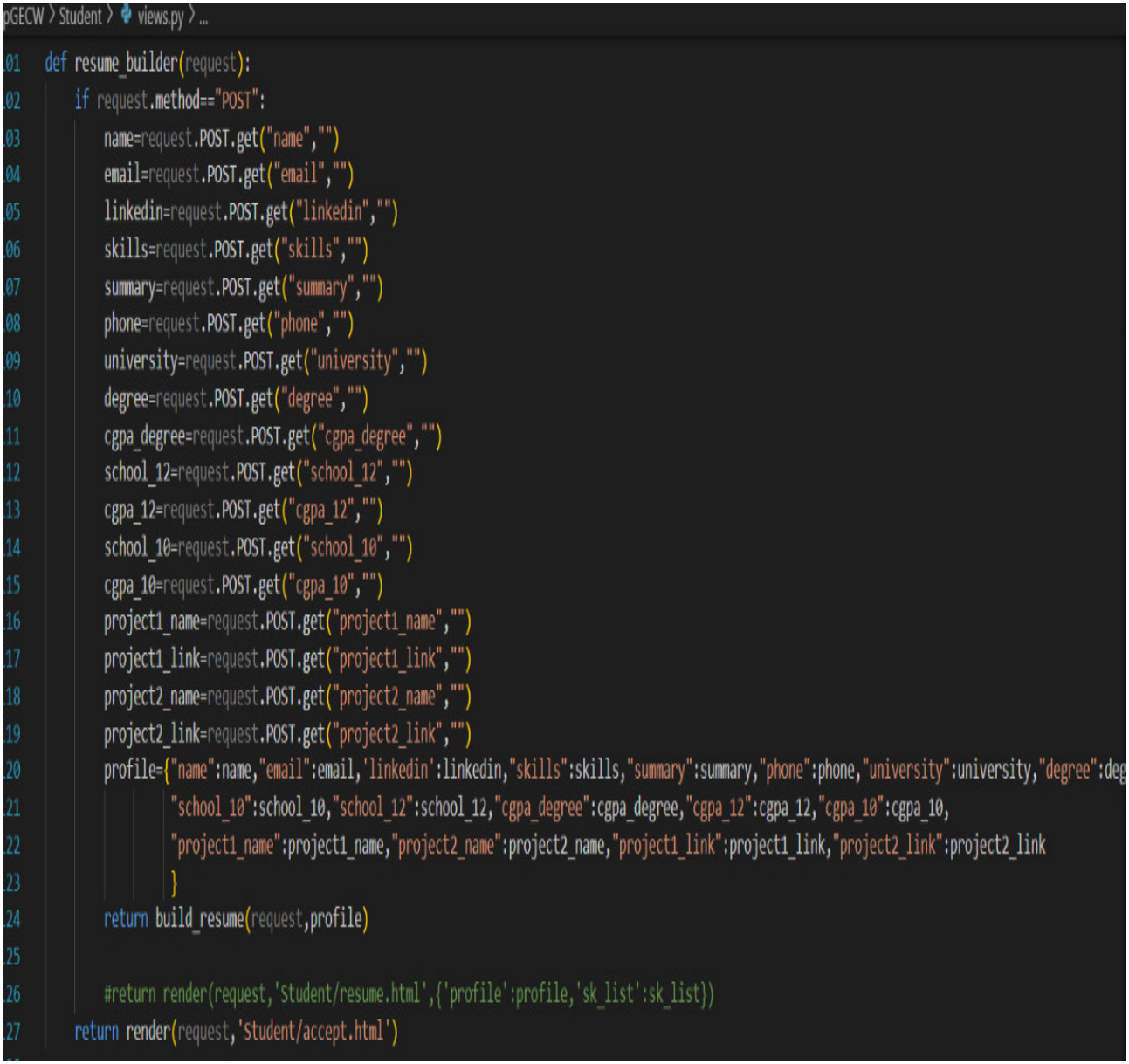


Figure 6.6: Resume builder module(1)

```
def build_resume(request,profile):
    skill_string=profile['skills']
    sk_list=skill_string.split(",")
    for s in sk_list:
        print(s)
    template=loader.get_template('Student/trial.html')
    html=template.render({'profile':profile,'sk_list':sk_list})
    options={
        'page-size':'Letter',
        'encoding':"UTF-8",
        'enable-local-file-access': "",
    }
    pdf=pdfkit.from_string(html,False,options)
    response=HttpResponse(pdf,content_type='application/pdf')
    response['Content-Disposition']='attachment'
    filename="resume.pdf"
    return response
```

Figure 6.7: Resume builder module(2)

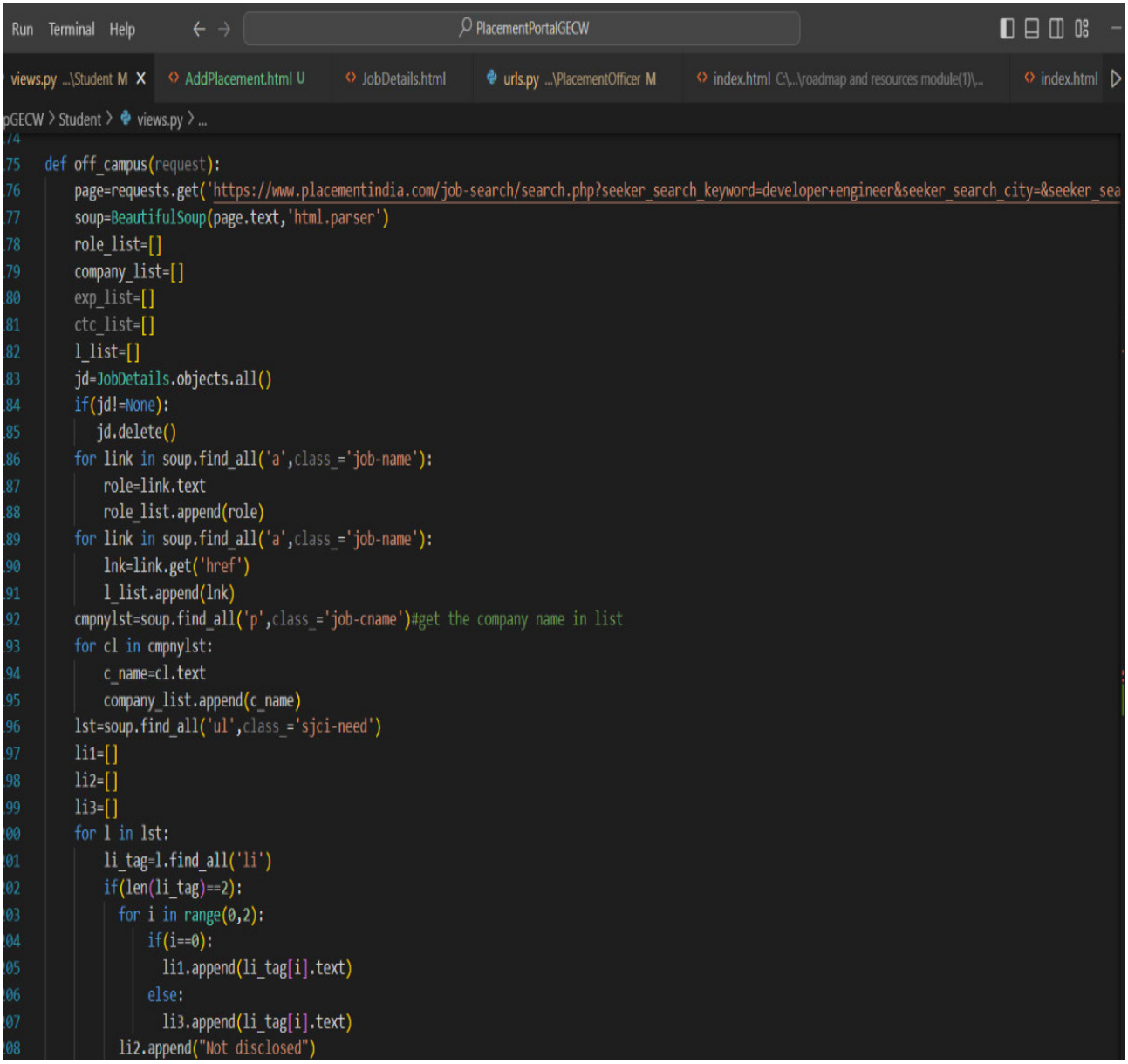


Figure 6.8: Off-campus placement module(1)

```python
def s_oncampus(request):
    jobs = OnCampusJobs.objects.all()
    return render(request, 'Student/StudentOnCampus.html', {'jobs': jobs})

def job_details(request,pk):
    job = get_object_or_404(OnCampusJobs, pk=pk)

    if request.method == 'POST':
        # Assuming you have the current student logged in and stored in `request.user.profile`
        student = request.user.profile
        # Create a new StudentOnCampusJobs object
        student_job = StudentOnCampusJobs(company_id=job, student_id=student)
        student_job.save()
        return redirect('s_oncampus')
    return render(request, 'Student/JobDetails.html', {'job': job})
```

Figure 6.9: On-campus placement module(1)

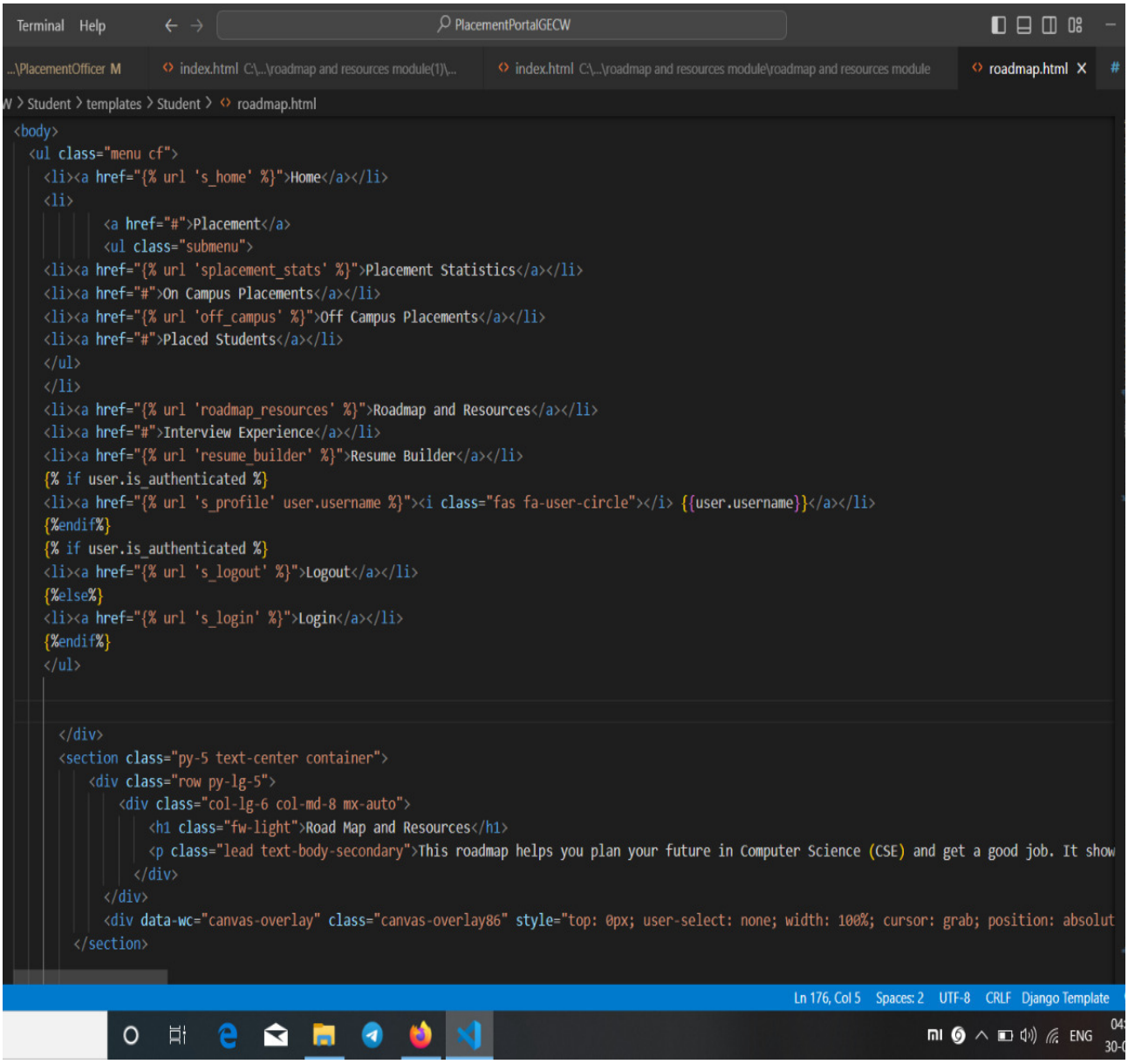


Figure 6.10: Roadmap and resources module(1)

6.3 SUMMARY

All functionalities have been successfully implemented.All modules are demonstrated and validated to ensure proper working.

CHAPTER 7
RESULTS AND DISCUSSION

7.1 INTRODUCTION

This section aims to address the results and discussions of the placement portal for GECW project. This project the need to automate and modernize the placement process, emphasizing several key objectives to be achieved. Firstly, there is a focus on enabling real-time updates, which suggests the importance of timely information dissemination regarding job opportunities, application statuses, and placement-related events. By providing up-to-date information in real-time, students and recruiters can stay informed and engaged throughout the placement cycle.

Efficient communication between students and recruiters is highlighted as another crucial aspect of the problem statement. This implies the need for a streamlined communication channel that facilitates interactions such as job applications, interview scheduling, feedback exchange, and general inquiries. An efficient communication system would foster better collaboration between students and recruiters, leading to smoother placement procedures.

Comprehensive data management emerges as a significant requirement, indicating the necessity to organize and manage large volumes of placement-related data effectively. This encompasses various aspects such as student profiles, job listings, recruiter information, application records, and placement progress tracking. A robust data management system would centralize all relevant information, making it easily accessible and searchable for both students and recruiters.

Improved tracking of placement progress is identified as a key goal, suggesting the need for mechanisms to monitor and evaluate the placement process's effectiveness. This involves tracking metrics such as placement rates, job acceptance rates, offer-to-acceptance ratios, and feedback loops. By analyzing these metrics, stakeholders can identify areas for improvement and make data-driven decisions to enhance the overall placement experience.

Ultimately, the overarching objective of the problem statement is to enhance the overall effectiveness and transparency of the placement experience. This entails providing students with more opportunities and a clear path to achieve them, while also ensuring that recruiters have access to a diverse pool of qualified candidates. By addressing these objectives, the proposed solution aims to revolutionize the traditional placement process, making it more efficient, transparent, and beneficial for all stakeholders involved.

7.2 RESULTS

Home Page

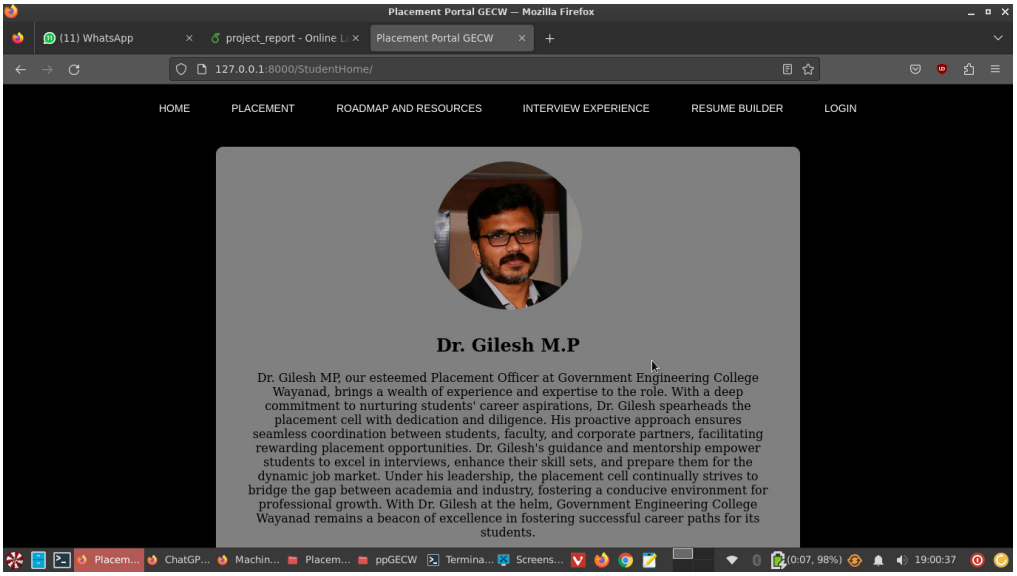


Figure 7.1: Home page

Student profile

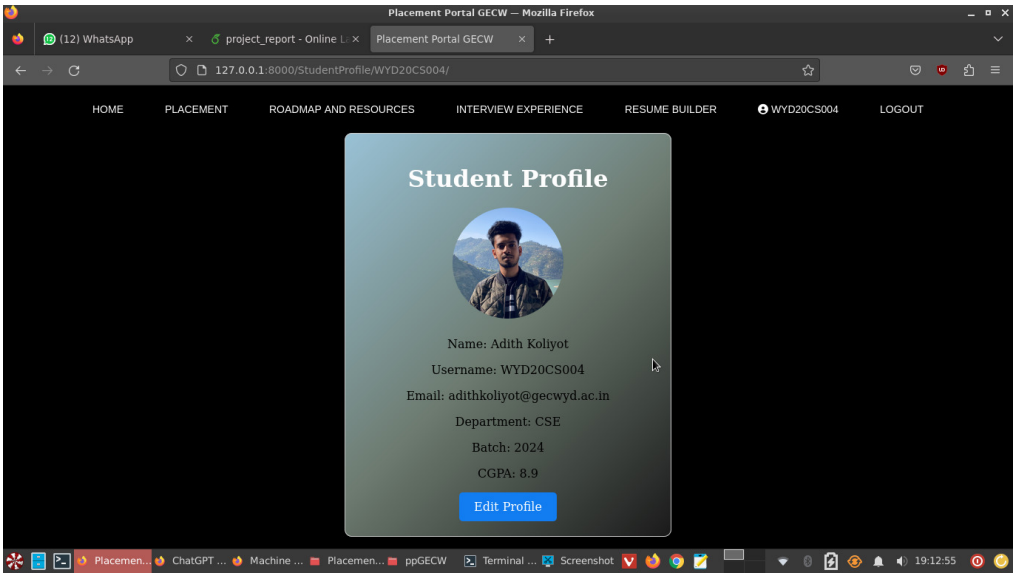


Figure 7.2: Student profile

Authentication

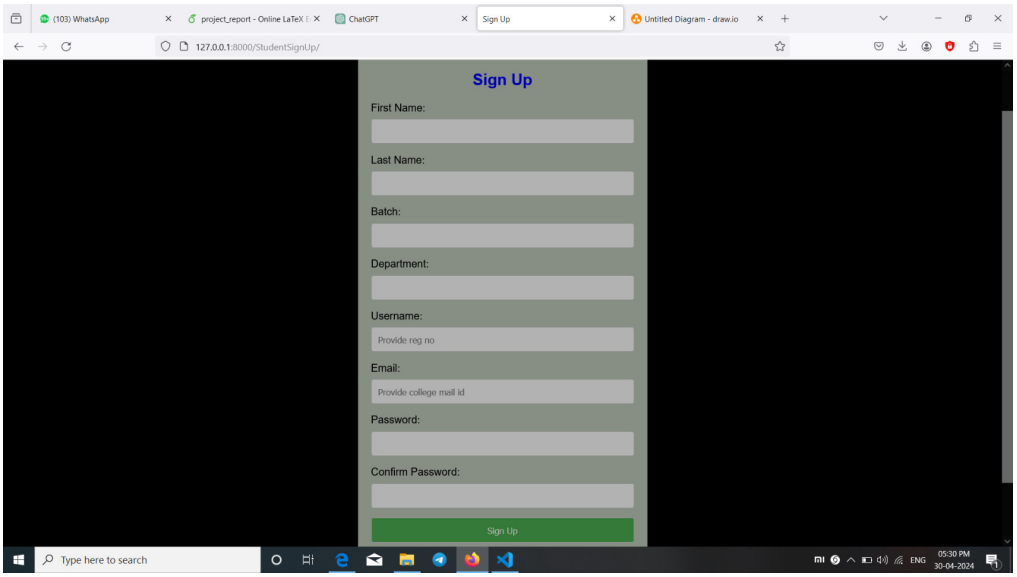


Figure 7.3: sign up for student

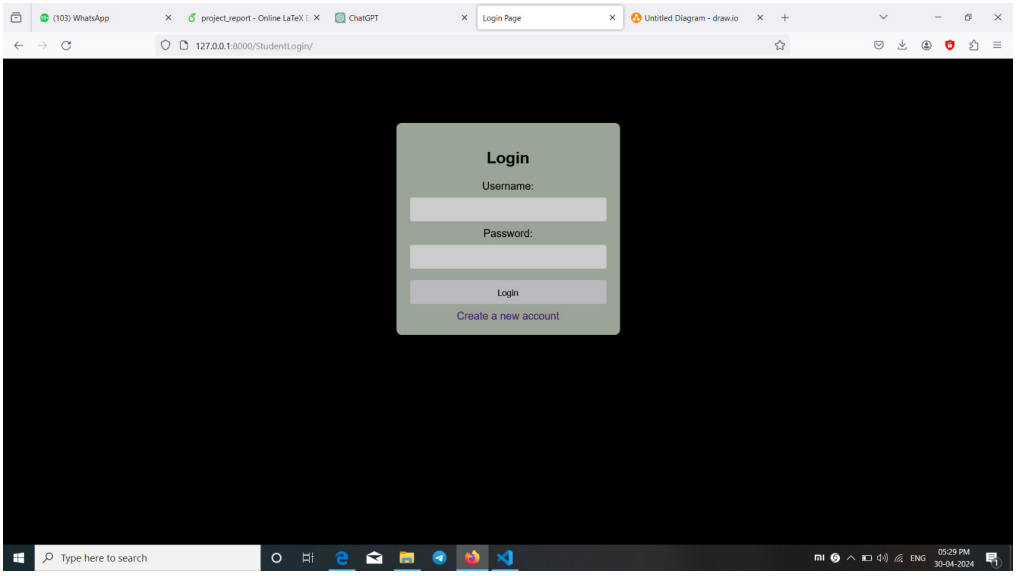


Figure 7.4: login for student

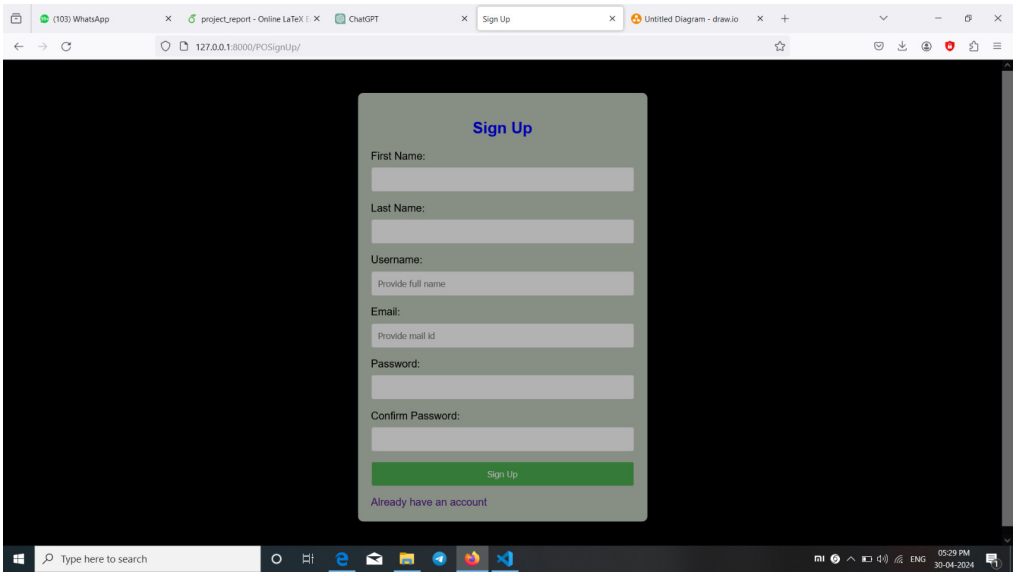


Figure 7.5: sign up for placement officer

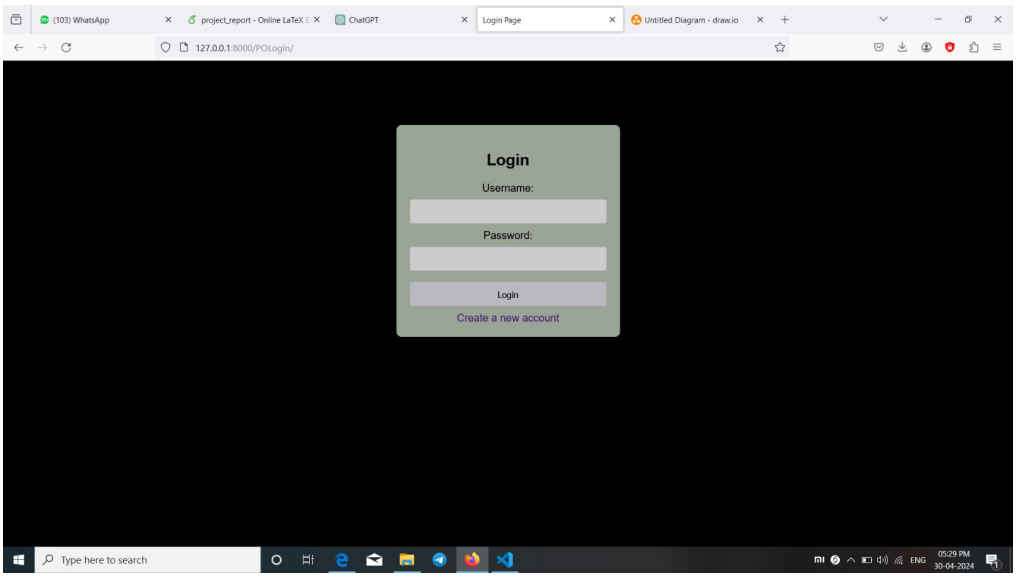


Figure 7.6: login for placement officer

On campus placement module

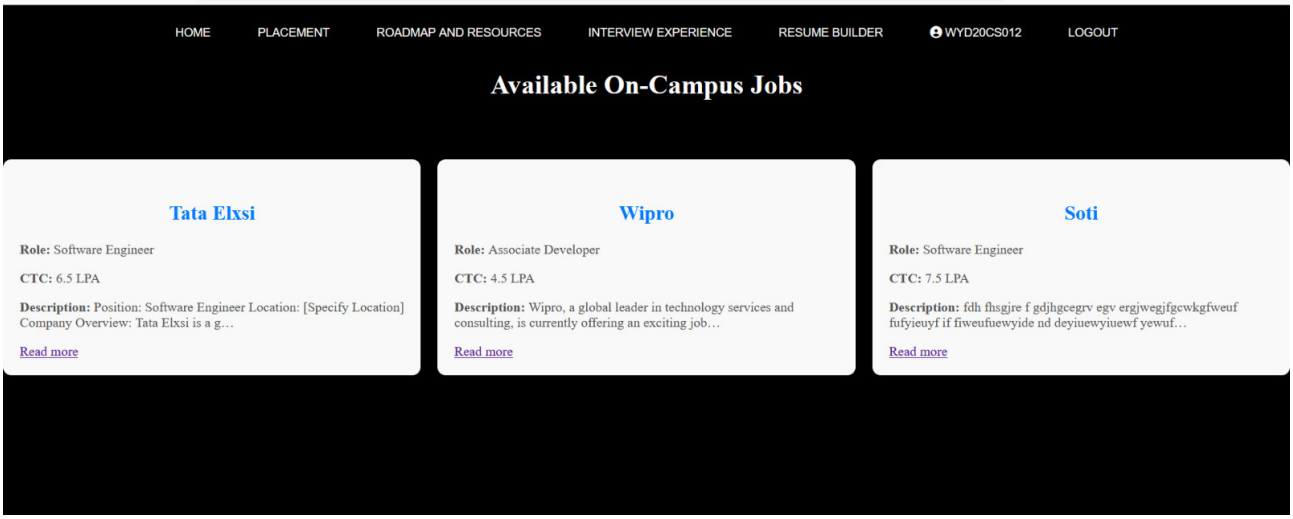


Figure 7.7: On campus job home page

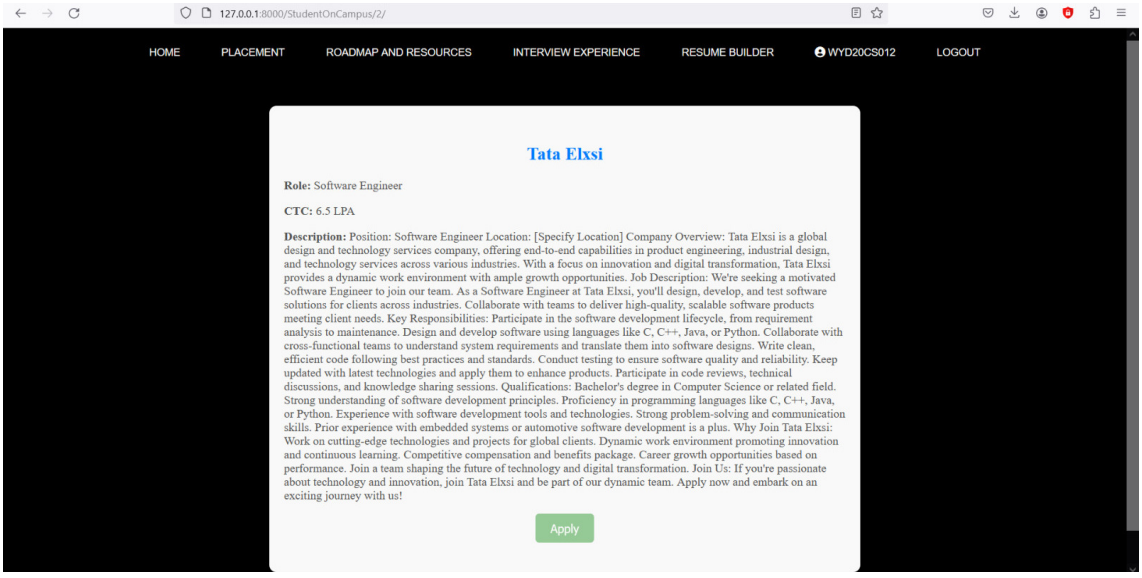


Figure 7.8: On campus job details page

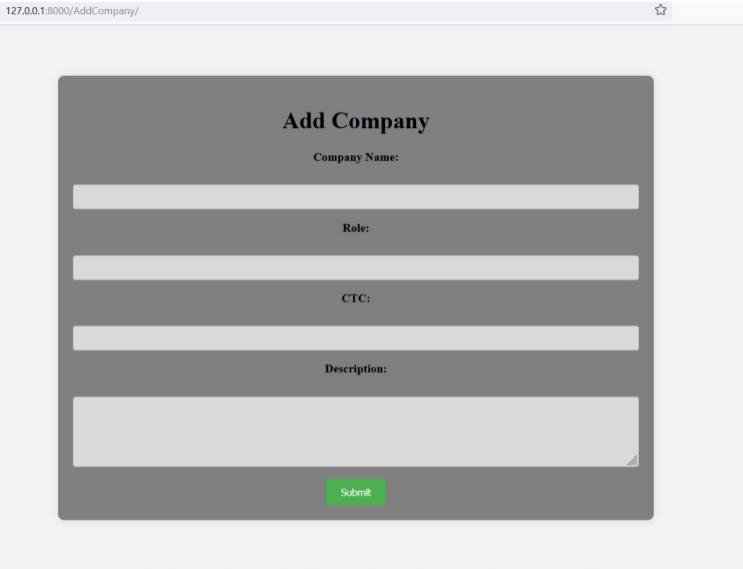


Figure 7.9: placement officer input page

Student Details

Reg. No	Name	Email	CGPA	Batch
WYD20CS012	Amaan Zain	zainamaan123@gmail.com	7.5	2024
WYD20CS045	Nived Narayanan KK	nived_20b158cs@gecwyd.ac.in	7.32	2024
WYD20CS016	Anu Antony	anu123@gmail.com	None	2024

Figure 7.10: Applied strudent details viewing page

Off campus placement module

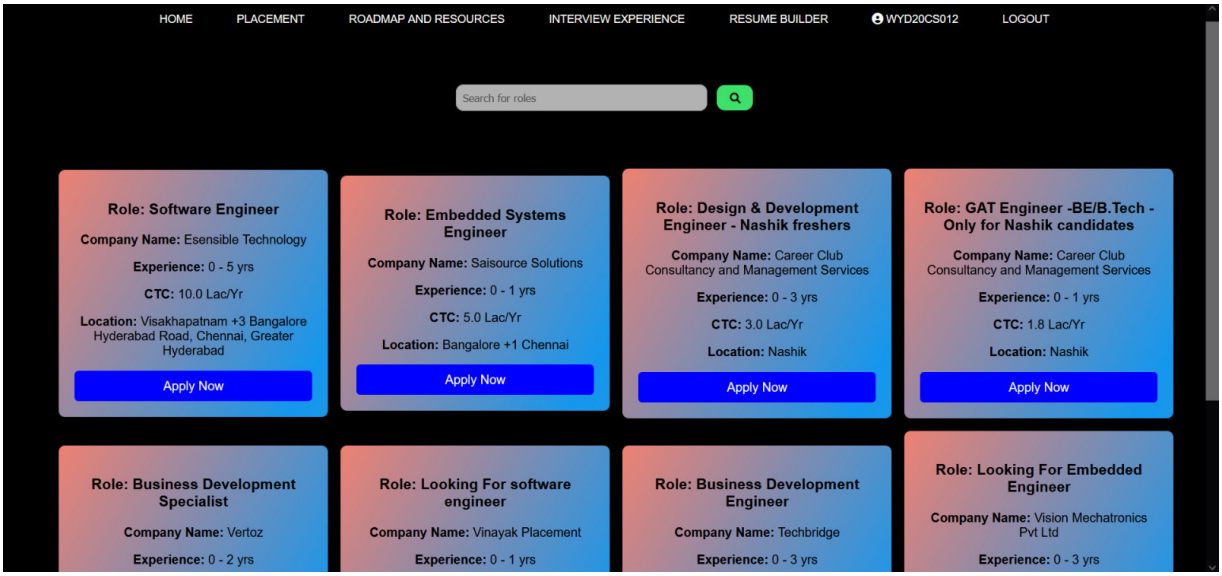
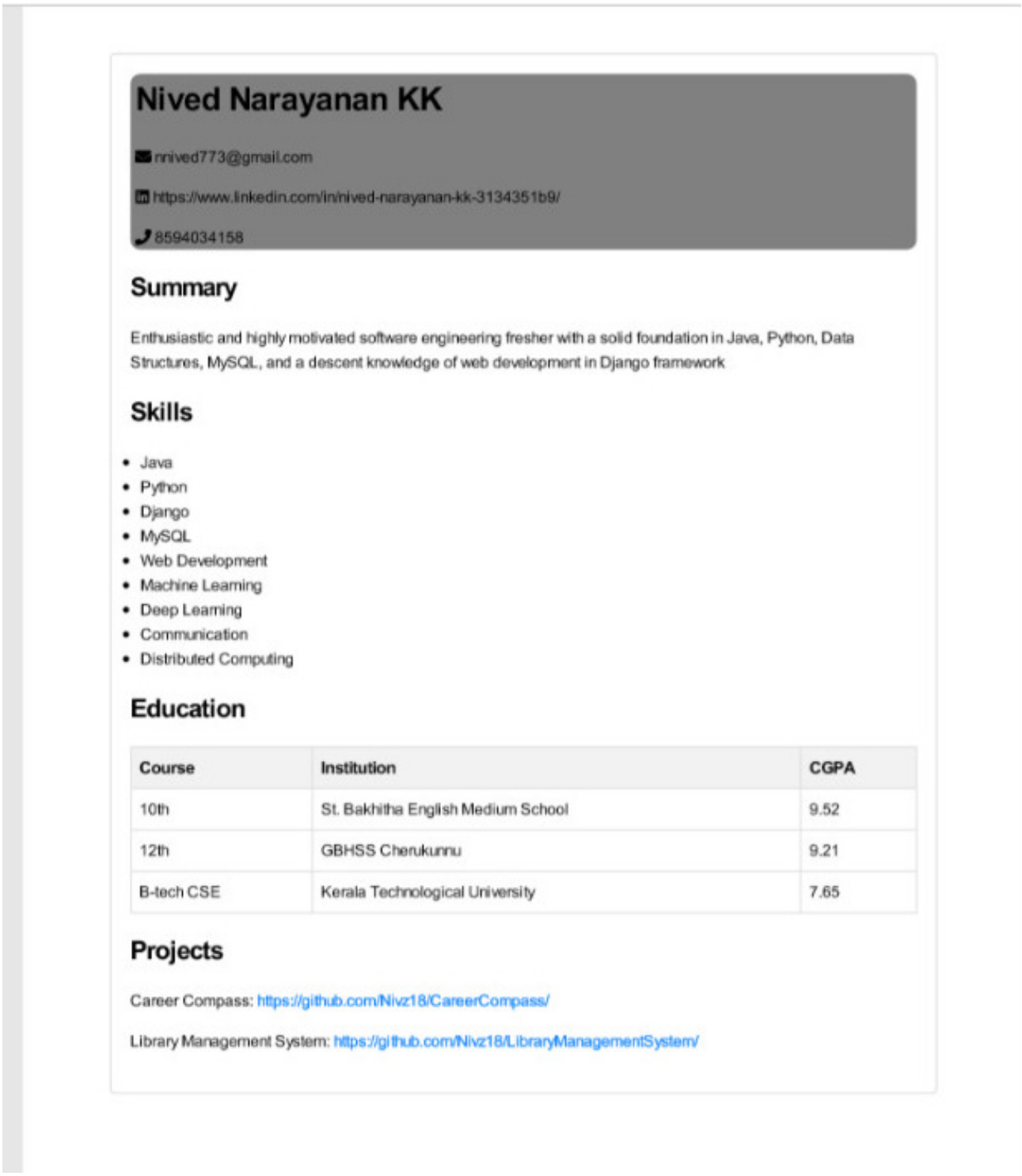


Figure 7.11: Off campus job home page

Figure 7.12: Placement statistics viewing page

Resume builder module

Figure 7.13: Resume input form(1)



Nived Narayanan KK

nnived773@gmail.com

https://www.linkedin.com/in/nived-narayanan-kk-3134351b9/

8594034158

Summary

Enthusiastic and highly motivated software engineering fresher with a solid foundation in Java, Python, Data Structures, MySQL, and a descent knowledge of web development in Django framework

Skills

- Java
- Python
- Django
- MySQL
- Web Development
- Machine Learning
- Deep Learning
- Communication
- Distributed Computing

Education

Course	Institution	CGPA
10th	St. Bakhitha English Medium School	9.52
12th	GBHSS Cherukunnu	9.21
B-tech CSE	Kerala Technological University	7.65

Projects

Career Compass: https://github.com/Nivz18/CareerCompass/

Library Management System: https://github.com/Nivz18/LibraryManagementSystem/

Figure 7.14: Generated resume

Roadmap and resources module

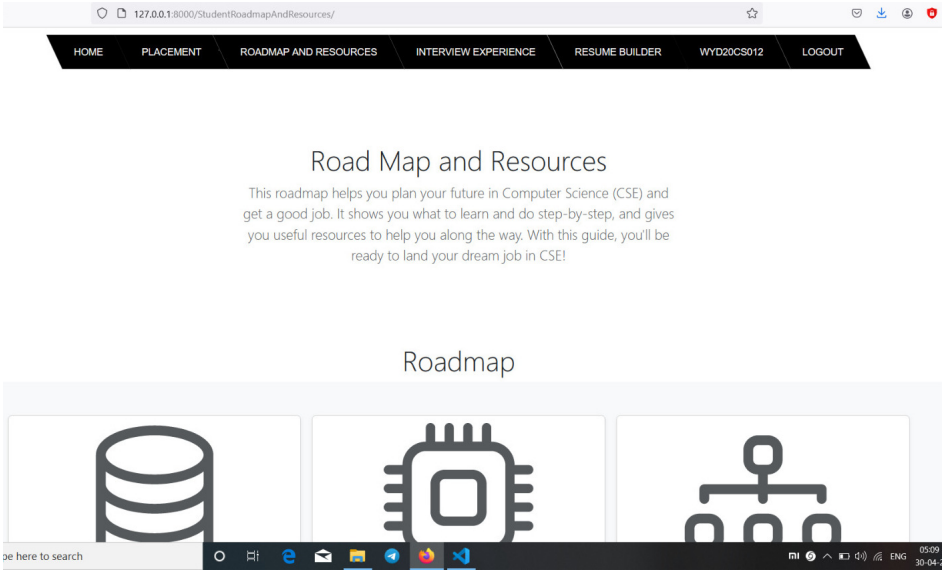


Figure 7.15: Roadmap and resources page(1)

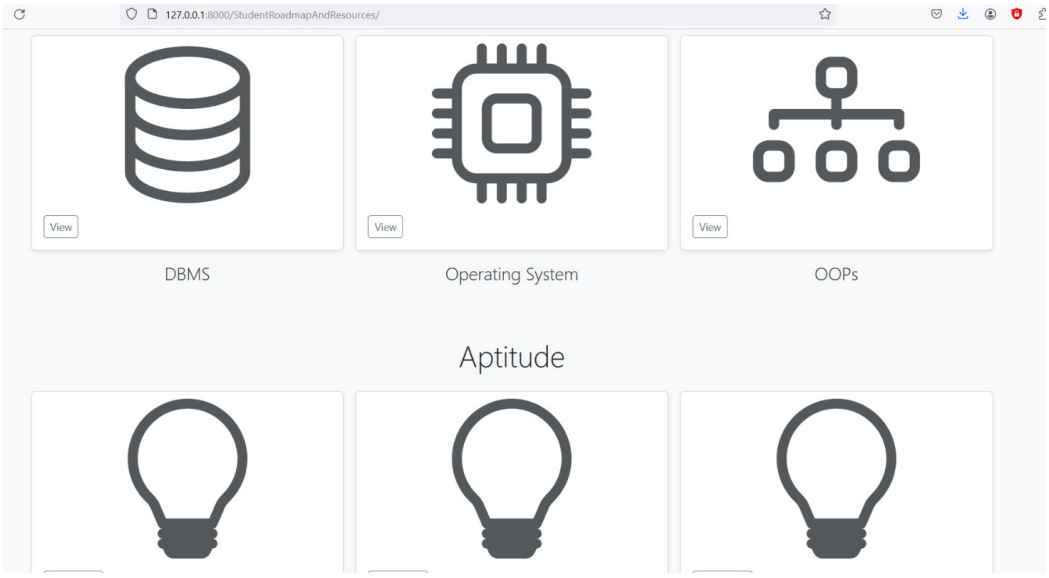


Figure 7.16: Roadmap and resources page(2)

CHAPTER 8
DOCUMENTATION

8.1 INTRODUCTION

Given the challenges presented by today's competitive job market, students often encounter difficulties in securing job placements due to a lack of sufficient guidance, clear plans, and available opportunities. The current job search process is characterized by disorganization, requiring students to independently search for jobs, determine their career trajectory, and locate resources. Furthermore, the manual handling of placement tasks results in delays, communication errors, and challenges in interview scheduling and result tracking, ultimately compromising the effectiveness of the placement process and leading to missed opportunities and miscommunication among stakeholders. Moreover, the absence of a structured analytics and reporting system poses difficulties for administrators in deriving actionable insights from placement data, impeding informed decision-making for continuous process improvement.

To address these issues, our initiative involves the creation of a user-friendly website that consolidates guidance, career plans, and resources into a single, accessible platform. By centralizing these essential elements, our aim is to streamline the job-hunting process for students, making it easier for them to identify and pursue job opportunities successfully. Ultimately, our website seeks to simplify the job search journey and provide students with the necessary support to navigate the complexities of the job market with confidence.

8.2 WORKING WITH THE PRODUCT

8.2.1 User characteristics

The main users of the system are students who studying in government engineering college Wayanad who uses the system to manage their placement activities. The placement officer of GEC Wayanad is the also a user who manage and control on-campus placement details. The Alumni are also users of the system . The users are required to have an Android phone with internet connectivity or a general computer system with internet connection to manage their user details and job applications. Mostly the users are students of GEC Wayanad who needs placement assistance and training.

8.2.2 Product functions

(i) User authentication

A login screen is provided in the web application which allows only authorized users to access the system.

(ii) On-campus placements details

On-campus placement offers are provided by the placement officer are listed and students can apply through the application.

(iii) Off-campus placements details

Off-campus placement offers integrated to our project from other external website using web scrapping.

(iv) Resume Building

A robust resume builder, empowering students to craft compelling resumes that effectively showcase their skills and experiences to potential employers.

(v) Roadmap and resources

It provides a long list of organized modules about the subjects which are necessary for placement and also some checkpoints in which need to be completed.

(vi) Placement Statistics

It display placement statistics year wisely and batch wisely for analyse the placement progress of the entire college.

(vii) Alumni Interview experience

Interview experience and placement experience of the alumni are available for the current students.

8.3 CONTACT

For more information contact:

- Nived Narayanan K K - nnived773@gmail.com
- Amaaan Zain N - zainamaanaz@gmail.com
- Anu Antony - antonyanu275@gmail.com
- Adith Koliyot - koliyotadith4@gmail.com

CHAPTER 9
CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

The computer sector are rapidly changing with high speed and most of the people are in competition to get a job in this market. It is certain that without correct training and guidance it is almost impossible to get the job. We proposed a custom made web application to efficiently manage and coordinate the placements of students in GEC Wayanad. Our system provides options to manage on campus placement as well as off campus placements. The students can make their resume within our website. It also provides roadmap and resources which is helpful for the student to prepare for their job interview and other placement rounds. It is basically a all in one system which can manage all the placements activities in a single window.

9.2 ADVANTAGES

- Manage of on-campus placements more efficiently.
- Off-campus offers are directly listed in the website.
- Resume can be build in a single click.
- Roadmap and resources are available which opens a gateway to how to prepare.

9.3 LIMITATIONS

- On-campus placement details need to be manually entered by placement officer.
- Roadmap and resources are static and it need to be updated manually.

9.4 FUTURE EXPANSION

There are a lot of future developments that can be done on the project. Additional functionalities can be provided to the application such as video chat with Alumni and current students which gives the students a real time experience. A news feed about the tech companies and MNCs can also be provided. The college placement cells real time notifications and data can be provided with the student

A mobile version of the web application can be developed and used. A placement assistance chat bot can be integrated with our system for better placement assistance. Such a system will improve the scalability and adaptability of the system.

REFERENCES

1. **S. Bikhchandani** (1997). Naukri.com. URL https://www.naukri.com/.

2. **R. Hoffman** (2002). LINKEDIN. URL https://www.linkedin.com/.

3. **A. Holovaty** and **S. Willison** (2005). Django. URL https://www.djangoproject.com/.

4. **J. L. S. L. Ivan Zhao, Chris Prucha** and **T. Schachman** (2013). notion.so. URL https://www.notion.so/.

5. **S. Jain** (2008). Geeks For Geeks. URL https://www.geeksforgeeks.org/.

6. **R. H. Tim Besse** (2007). Glassdoor. URL https://www.glassdoor.co.in.

7. **S. Verma** (1998). Placement India. URL https://www.placementindia.com/.