

A
PROJECT REPORT
ON
ONLINE JOB PORTAL
and
WEB SCRAPING

Submitted by

BHATHGARA SMIT JITENDRA (ET21MTCA004)

Under the Guidance Of
Prof. Zankhana Vaishnav

In fulfillment of the award of the degree

of

Master of Computer Application

At

Sarvajanik College of Engineering & Technology, Surat



Sarvajanik University, Surat

May, 2023



SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
MASTER OF COMPUTER APPLICATION
Academic Year 2022-23



Date: 06-05-2023

This is to certify that the project entitled "Online Job Portal and Web Scraping" has been Submitted by BHATHGARA SMIT JITENDRA (ET21MTCA004) towards the fulfillment of the degree of Master of Computer Applications (M.C.A.) in (4th Semester) of Sarvajanik University, Surat during the academic year 2022-23.

Guide Name: Prof. Zankhana Vaishnav

(Guide's Signature)

**(Prof. Prashant Keshwani
Head of M.C.A. Department)**

Examiners Signature:

1. _____

2. _____

3. _____



Reference No: LBU23May010

Date - 2ndMay,2023

TO WHOM IT MAY CONCERN

This is to certify that **Mr. Smit Jitendra Bhathagara (ET21MTCA004)** a student of **Logix Built Infotech** has successfully completed 04 (Four) Months (from 1stJan,2023 till 2ndMay,2023) long internship program at **Logix Built Uptech**. He is working on Project -“**online job portal**” & “**web scraping**” under the guidance of **Mr. Ashish S Mishra**.

During this tenure with us, we have found him sincere and diligent in his training. He was responsible, hardworking, and committed to his work. His contributions are valuable to our organization.

We wish him all the best in his future endeavors and hope that he will have a very successful career.

Thank you.
Yours faithfully,
For **Logix Built Uptech**

Siddharth Pandya
(CEO)



Reference No: LBU23Apr009

Date - 2ndMay,2023

To

MCA Department,
Sarvajanik College Of Engineering & Technology

Subject :- No Source code & Database for "**online job portal**"

This is to inform you that your college student named **Smit Jitendra Bhathagara** is working on a project name "**online job portal**" for our company. As per our company policy we will not deliver the source code of the project please contact us if you have any question regarding the same.

We have found him sincere and diligent in their duties and responsibilities. He is responsible, hardworking and committed to work.

Thank you.
Yours faithfully,
For **Logix Built UpTech**

Chirag Patel
(Sr.Project Manager)



Reference No: LBU23Apr009

Date - 2ndMay,2023

To

MCA Department,
Sarvajanik College Of Engineering & Technology

Subject :- No Source code & Database for "**web scraping**"

This is to inform you that your college student named **Smit Jitendra Bhathagara** is working on a project name "**web scraping**" for our company. As per our company policy we will not deliver the source code of the project please contact us if you have any question regarding the same.

We have found him sincere and diligent in their duties and responsibilities. He is responsible, hardworking and committed to work.

Thank you.
Yours faithfully,
For **Logix Built UpTech**

Chirag Patel
(Sr.Project Manager)

Self Declaration

Title of the Project: ONLINE JOB PORTAL And WEB SCRAPING

Enrolment Number	Student Name
ET21MTCA004	Bhathgara Smit Jitendra

I hereby declare that the above-mentioned project report submitted by me/us has been prepared by me/us and is original in its content and it has not been submitted anywhere else.

I confirm that the report is only prepared for academic requirements, not for any other purpose. It might not be used by anyone for any other purpose.

Student's Signature:

(Bhathgara Smit Jitendra)

ACKNOWLEDGEMENT

I express my sense of pleasure towards the Training Officer of LogixBuilt Infotech who gave me a chance to do the project work. He always motivates all the trainees and provides extraordinary infrastructure and resources to work. Moreover, I learned the lesson of “commitment to work” from him. I memorize him for his cordial and gentle nature.

I express our deep sense of gratitude and indebtedness to my guide Mr. Ashish Mishra for accepting me to work under his training and supervision. He took a prolonged interest in my work and directed me toward the predefined goal. He has shown me a way to pursue excellence. He witnessed our work every time and helped a lot. He has been a big factor of motivation in my project.

I am also very much thankful to Prof. Zankhana Vaishnav and all other Faculties of the MCA Department, Sarvajanik College of Engineering & Technology (SCET) for the academic advancement and for providing an opportunity for project development training.

I sincerely thank Prof. Prashant Keswani, Head of the MCA department, who provide me a constant motivation for knowledge acquisition and moral support during project work. I would once again like to express my heartiest gratitude to my family and friends who have always guided me toward the path of success and helped to make this project work successfully.

Student's Signature:

(Bhathgara Smit Jitendra)

Online Job Portal

Index

Sr No.	INDEX	Page No.
1.	Introduction 1.1 Existing System 1.2 Need for the New System 1.3 Objective of the New System 1.4 Problem Definition 1.5 Core Components 1.6 Project Profile 1.7 Assumptions and Constraints 1.8 Advantages and Limitations of the Proposed System	2 3 4 5 6 7 8 9 10
2.	Requirement determination & Analysis 2.1 Requirement Determination 2.2 Targeted Users	11 12 13
3.	System Design 3.1 Use Case Diagram 3.2 Class Diagram 3.3 Interaction Diagram 3.4 Activity Diagram 3.5 Data Dictionary	14 15 16 17 18 23
4.	Development 4.1. Coding Standards 4.2. Screen Shots	25 26 27
5.	Agile Documentation 5.1 Agile Project Charter 5.2 Agile Roadmap / Schedule 5.3 Agile Project Plan 5.4 Agile User Story (Minimum 3 Tasks) 5.5 Agile Release Plan 5.6 Agile Sprint Backlog 5.7 Agile Test Plan 5.8 Earned-value and burn charts	34 35 36 37 38 39 40 41 42
6.	Proposed Enhancements	43
7.	Conclusion	44
8.	Bibliography	45

Web Scraping

Index

Sr No.	INDEX	Page No.
1.	Introduction 1.2 Existing System 1.2 Need for the New System 1.3 Objective of the New System 1.4 Problem Definition 1.5 Core Components 1.6 Project Profile 1.7 Assumptions and Constraints 1.8 Advantages and Limitations of the Proposed System	47 48 49 50 51 52 53 54 55
2.	Requirement determination & Analysis 2.1 Requirement Determination 2.2 Targeted Users	56 57 58
3.	System Design 3.1 Use Case Diagram 3.2 Class Diagram 3.3 Interaction Diagram 3.4 Activity Diagram	59 60 61 62 63
4.	Development 4.1. Coding Standards 4.2. Screen Shots	64 65 66
5.	Proposed Enhancements	72
6.	Conclusion	73
7.	Bibliography	74

Project 1

Online Job Portal

1

Introduction

1.1. Existing System

- In the existing system, job seekers and employers have to rely on offline channels such as newspaper advertisements, recruitment agencies, or personal networks to find job opportunities or candidates. This process can be time-consuming, inefficient, and costly.
- Job seekers have to physically visit potential employers, submit resumes and attend interviews, which can be a hassle and limit their options. Similarly, employers have to manually sort through resumes and applications, which can be time-consuming and tedious.
- Furthermore, there is often a lack of transparency in the hiring process, as employers may not provide feedback or updates on the status of the application, which can be frustrating for job seekers.
- Overall, the existing system lacks convenience, efficiency, and transparency, and there is a need for a more accessible and user-friendly platform for job seekers and employers. This is where the proposed online job portal comes in, as it aims to address these issues and provide a better experience for users.

1.2. Need for the New System

1. The current job search process is challenging and time-consuming for job seekers and employers.
2. Job seekers have to navigate multiple websites and offline channels to find job openings, while employers have to invest significant resources in recruitment.
3. The current system lacks transparency and communication between employers and job seekers, leading to frustration and lost opportunities.
4. There is a need for a more efficient and user-friendly platform to centralize job listings and facilitate communication between employers and job seekers.
5. An online job portal can streamline the recruitment process, save time and money for both job seekers and employers and provide better access to job opportunities and candidates.

By addressing these needs, the new online job portal system developed using Django framework and SQLite database aims to overcome the limitations of the existing system and provide a more efficient, user-friendly, and effective platform for job-seekers and employers.

1.3. Objective of the New System

- Enhance job-seeker experience with advanced features such as user authentication, resume management and resume builder, job alerts.
- Streamline employer operations by simplifying job postings, application management, and resume builder.
- Ensure data security and privacy by implementing robust security measures.
- Provide enhanced administrative control for efficient system management.

By achieving these objectives, the new system aims to provide a user-friendly, efficient, and effective platform for job-seekers and employers, ultimately facilitating a seamless and productive job search process.

1.4. Problem Definition

- Limited Access: The existing system may have limited access to job opportunities beyond geographical boundaries, limiting the job-seekers' options.
- Lack of Advanced Features: The existing system may lack advanced features such as user authentication, resume management which could enhance the user experience.
- Security and Privacy Concerns: The existing system may not have robust security measures in place to protect user data, leading to potential data breaches or unauthorized access.
- Complex Administrative Management: The existing system may lack efficient administrative controls for managing job postings, applications, and user profiles, resulting in challenges for system administrators.
- Communication and Coordination Issues: The existing system may face challenges in facilitating smooth communication and coordination between job-seekers and employers, leading to delays and misunderstandings.

The new online job portal system aims to address these problems and provide a more efficient, user-friendly, and effective platform for job-seekers and employers, offering advanced features, enhanced security, and streamlined processes.

1.5. Core Components

Frontend	HTML , CSS
Backend	Python - Django
Database	SQLite

1.6. Project Profile

Project Title	Online Job Portal
Organization	Logix Built InfoTech Surat.
Platform	Windows 10
Tech Stack	Django
Database	SQLite
Project Guide	Mr. Aashish Mishra
Submitted to	Sarvajanik College of Engineering & Technology
Presented by	Bhathgara Smit

1.7. Assumptions and Constraints

Assumptions:

- Users have basic knowledge of using web-based applications and are familiar with job search processes.
- Users have reliable internet connectivity and compatible web browsers to access the online job portal.
- Users will provide accurate and complete information while registering and using the system.
- The system will be tested and used in a controlled environment, and necessary security measures will be implemented to protect user data and prevent unauthorized access.

Constraints:

- Time and resources: The project is constrained by the available time and resources for development, testing, and deployment.
- Technology limitations: The project is limited by the capabilities and constraints of the Django framework and SQLite database.
- User interface: The user interface may be limited by the supported browsers and devices, and may not be optimized for all screen sizes and resolutions.
- Security: The project must adhere to security best practices, but it may be constrained by the limitations of the chosen technology stack and available resources.

1.8. Advantages and Limitations of the Proposed System

Advantages of the Proposed System:

- User-friendly Interface: The online job portal provides a user-friendly interface for job-seekers and employers, making it easy to navigate, search for jobs, and manage job applications.
- Efficient Job Matching: The system uses advanced algorithms for job matching, enabling job-seekers to find relevant job openings based on their qualifications and preferences, and helping employers to identify qualified candidates quickly.
- Administrative Controls: The system provides administrative controls for managing job postings, applications, user profiles, and system settings, enabling efficient system management and maintenance.

Limitations of the Proposed System:

- Technology Constraints: The system is constrained by the capabilities and limitations of the Django framework and SQLite database, which may affect the scalability, performance, and customization options of the system.
- User Connectivity: The system relies on reliable internet connectivity, which may be a limitation in areas with poor or unstable internet access.
- Device Compatibility: The system may have limitations in terms of compatibility with different browsers, devices, and screen sizes, which may affect the user experience.
- Security Considerations: While efforts have been made to implement security best practices, the system may still have limitations in terms of security vulnerabilities and risks, which need to be addressed to protect user data and prevent unauthorized access.

2

Requirement Determination & Analysis

2.1. Requirement Determination

- User Requirements: The system must cater to the needs of two types of users - job-seekers and employers. Job-seekers should be able to easily search for jobs, create and manage their profiles, upload resumes, and apply for jobs. Employers should be able to post job openings, search for resumes, and manage job applications.
- Functional Requirements: The system should include features such as user registration and login, job posting and search, resume management, and administrative controls for managing system settings, user profiles, and job postings.
- Performance Requirements: The system should be efficient and responsive, with fast load times, quick search results, and smooth navigation. It should be able to handle concurrent user requests and manage large databases of job-seekers and job openings.
- Security Requirements: The system should implement appropriate security measures, such as user authentication, data encryption, and protection against cross-site scripting (XSS) and SQL injection attacks, to safeguard user data and prevent unauthorized access.
- Usability Requirements: The system should be user-friendly, with a clear and intuitive interface, easy-to-use features, and helpful error messages. It should be accessible on different browsers, devices, and screen sizes, and comply with accessibility standards.
- Constraints: The project is constrained by the available time, resources, and technology stack (Django framework and SQLite database), which may impact the design, development, testing, and deployment of the system.

2.2. Targeted Users

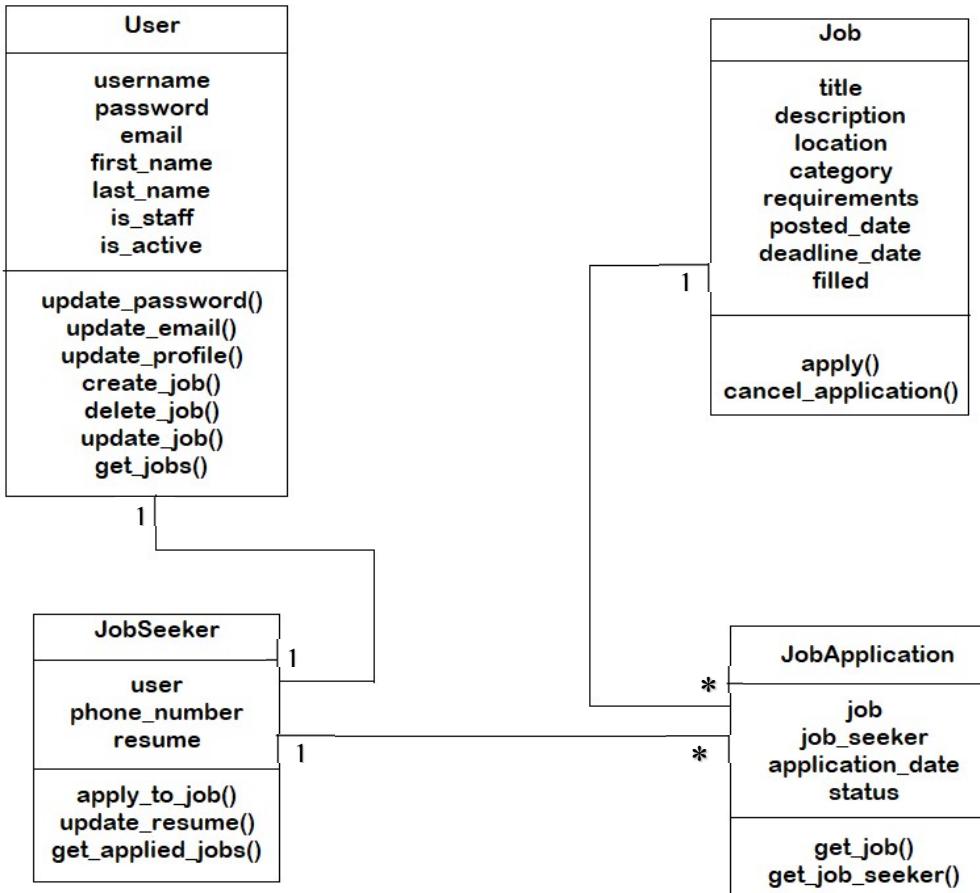
- The online job portal is designed to serve two main targeted users: job seekers and employers.
- Job seekers are individuals who are actively looking for employment opportunities. They may be recent graduates, individuals looking for a career change, or those who are currently employed but looking for new opportunities.
- Employers are businesses or organizations of all sizes who are looking to recruit qualified candidates for their job openings. They may be looking for full-time, part-time, or contract employees, and may have various job roles available.
- The job portal aims to provide a user-friendly platform for job seekers to browse and apply for job openings, manage their job applications and update their profiles. It also provides employers with a central platform to post job openings, manage applications, and communicate with job seekers.
- Overall, the job portal targets both job seekers and employers and aims to streamline the recruitment process and provide a more efficient and transparent platform for job search and hiring.

3
System Design

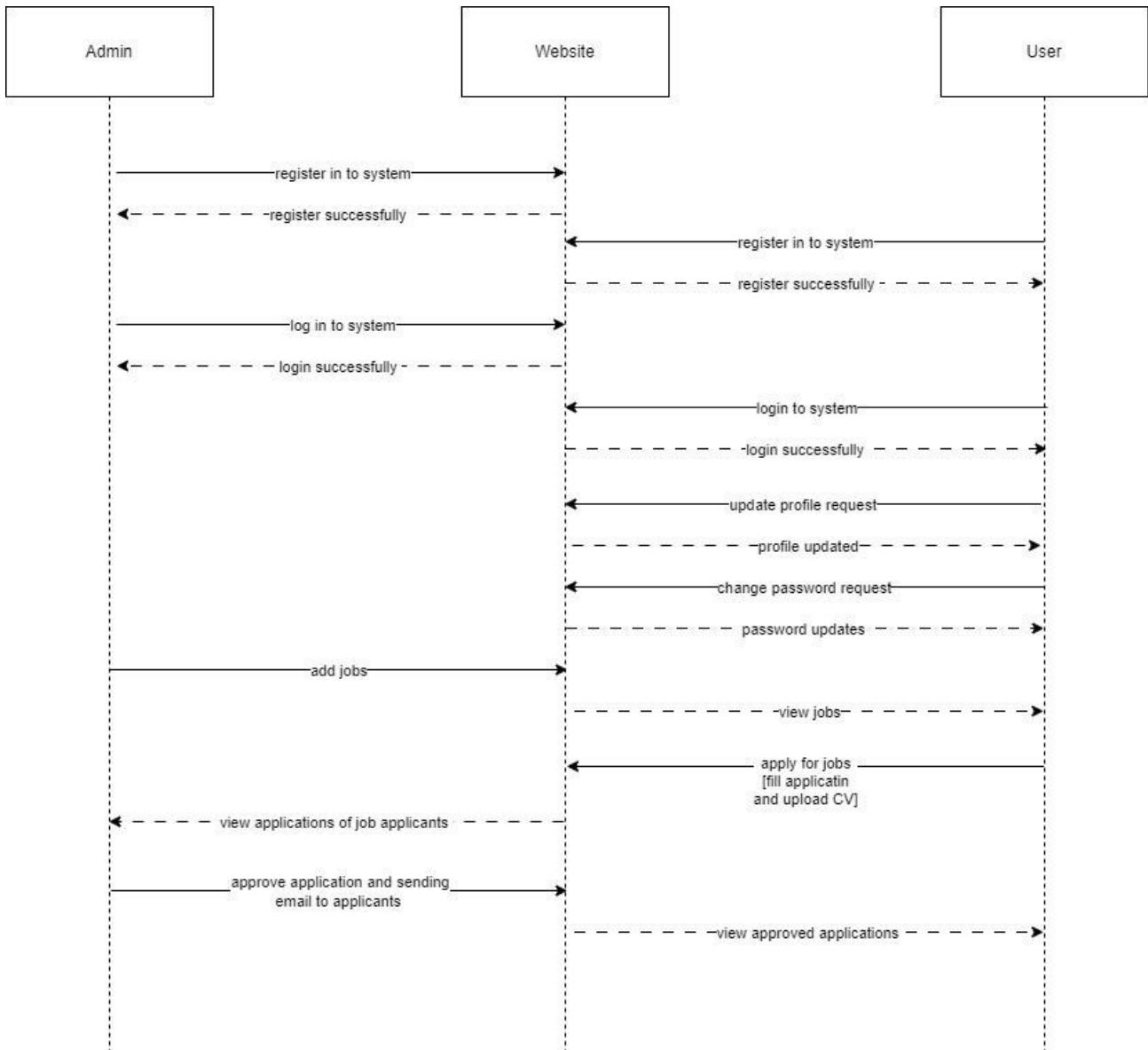
3.1. Use Case Diagram



3.2. Class Diagram

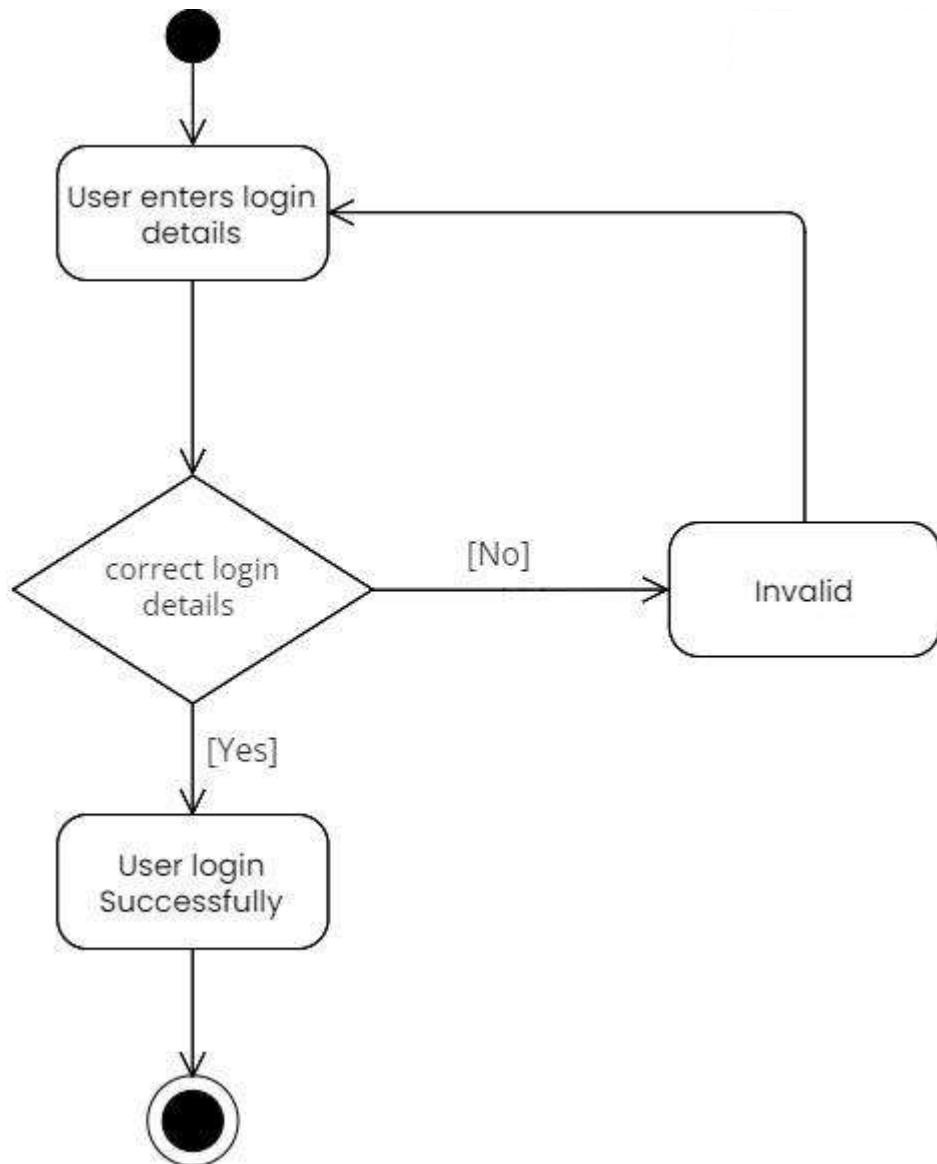


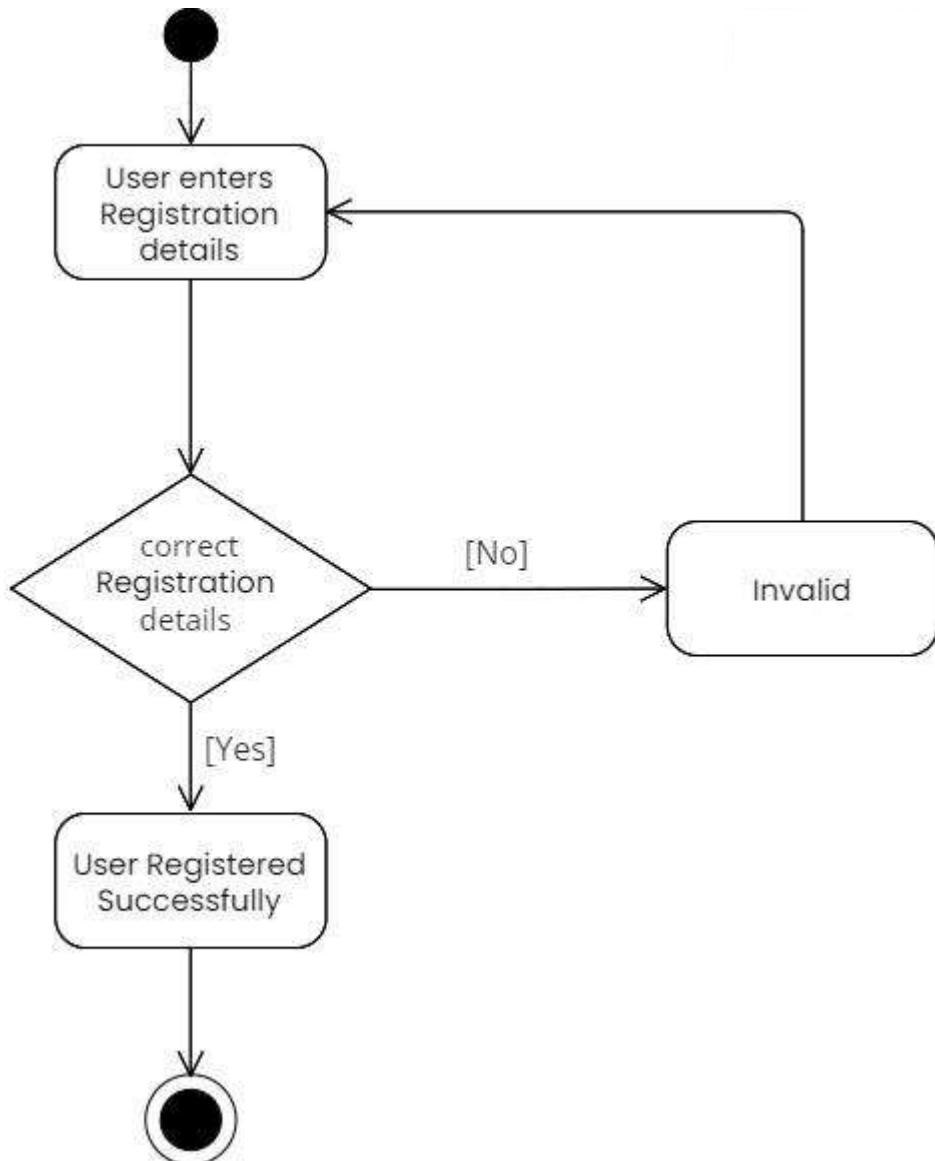
3.3. Interaction Diagram

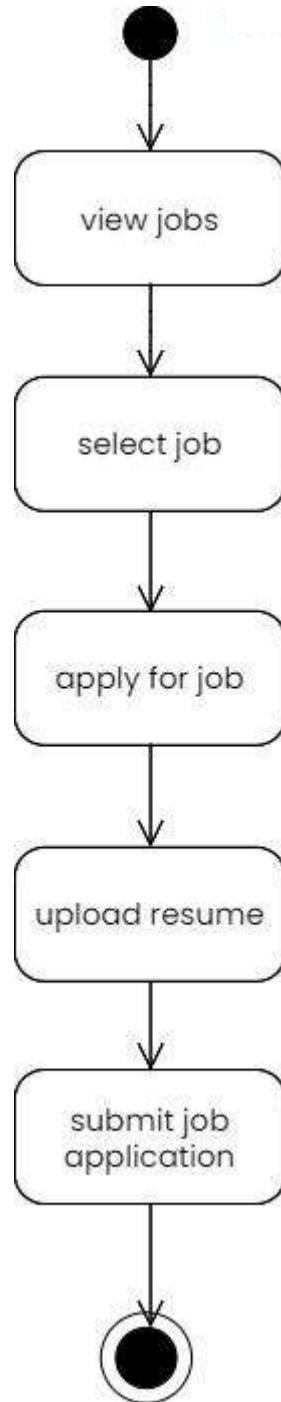


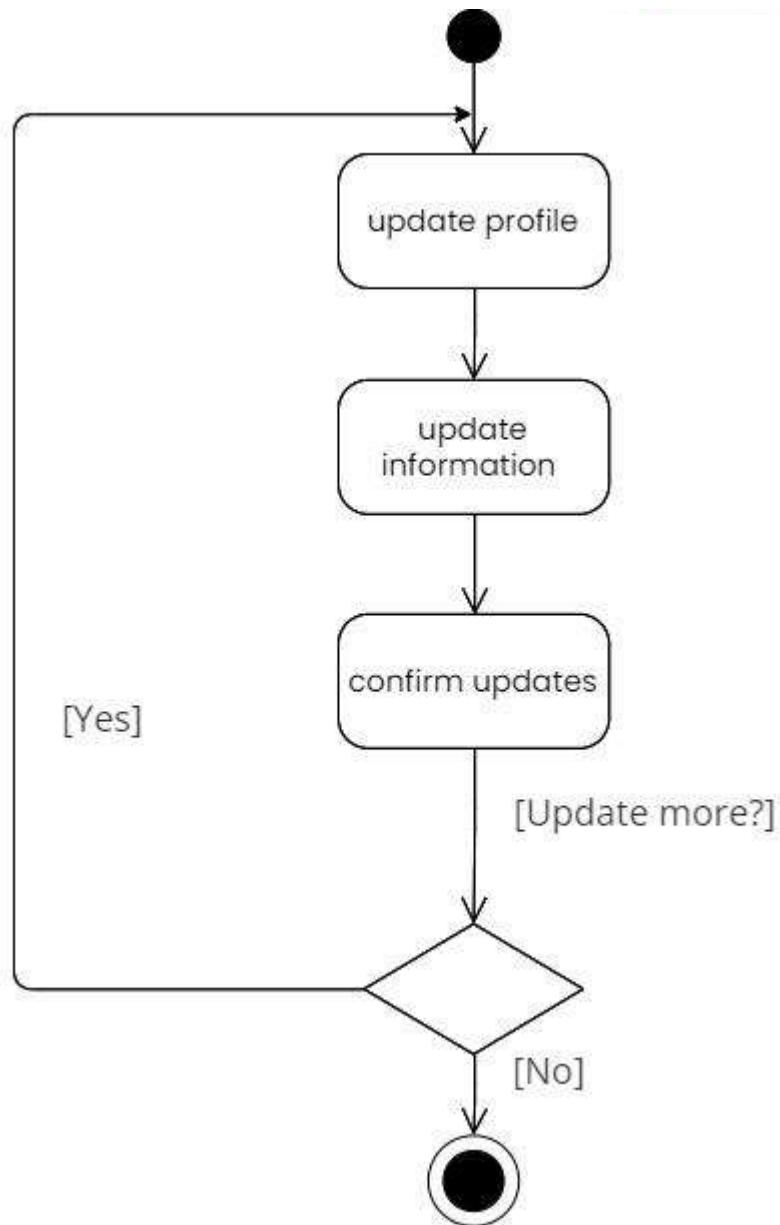
3.4. Activity Diagram

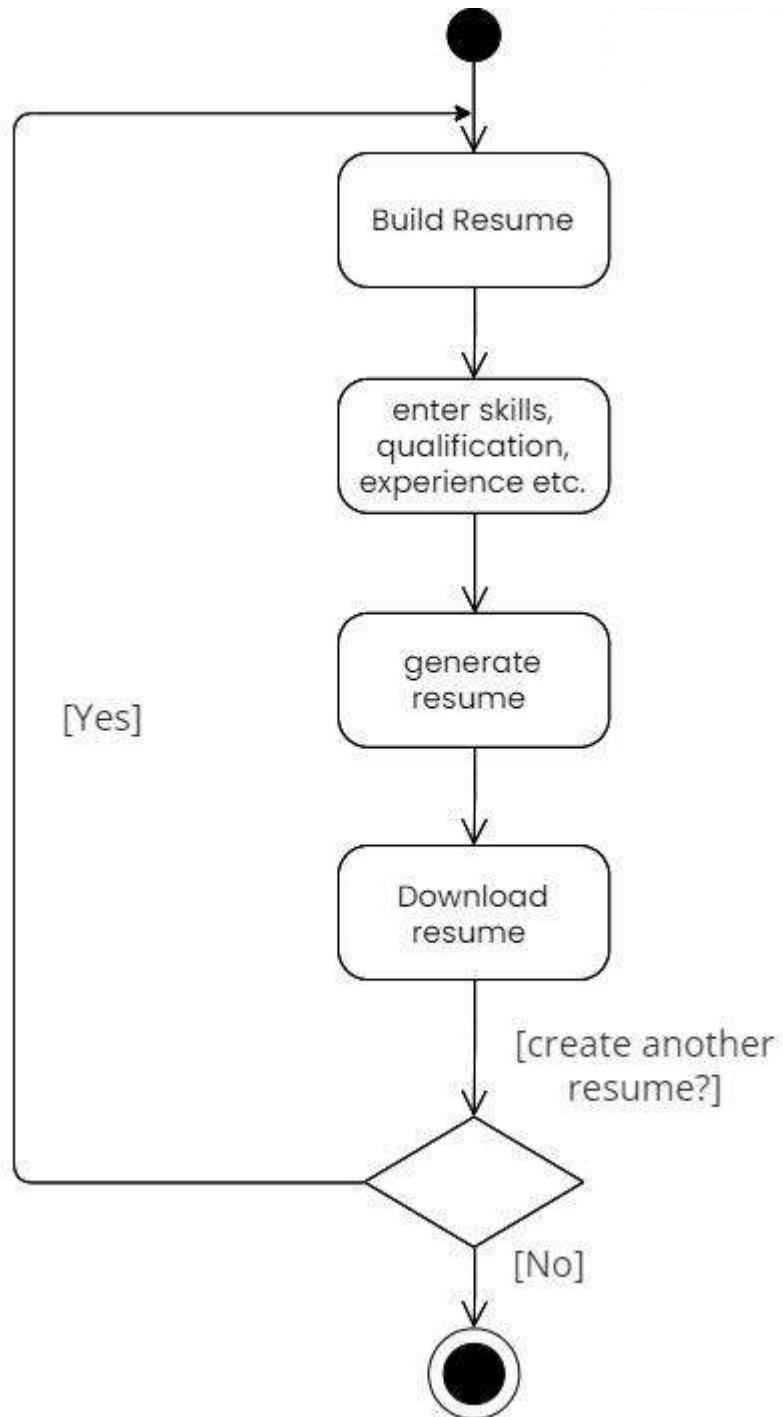
Login :



Registration :

Apply for jobs :

Update Profile :

Resume Builder :

3.5. Data Dictionary

Jobseeker Table

Sr. No	Field Name	Datatype (Size)	Description of field
1	Username	Foreign key	Jobname of job
2	email	Foreign key	Username of jobseeker
3	first_name	CharField	description of application
4	last_name	CharField	Application is approved or not
5	phone_number	CharField	Resume or CV
6	is_verified	BooleanField	Verify user
7	is_superuser	BooleanField	Clarify if user is admin

Jobs Table

Sr. No	Field Name	Datatype (Size)	Description of field
1	jobname	CharField	Jobname of job
2	company	CharField	company of job
3	description	CharField	description of job
4	pay	CharField	pay of job
5	address	CharField	address of job
6	application_start_date	DateField	application_start_date of job
7	application_end_date	DateField	application_end_date of job

Application Table

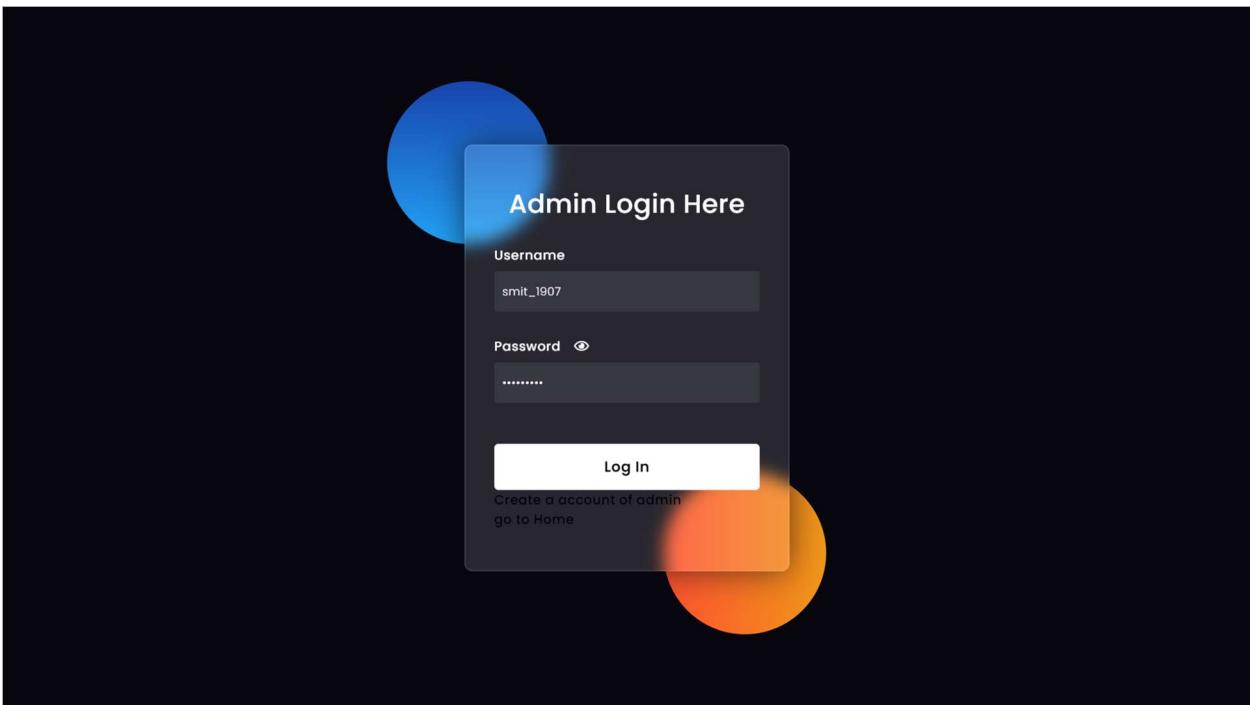
Sr. No	Field Name	Datatype (Size)	Description of field
1	job	Foreign key	Jobname of job
2	jobseeker	Foreign key	Username of jobseeker
3	desc	CharField	description of application
4	approve	BooleanField	Application is approved or not
5	file	FileField	Resume or CV

4
Development

4.1. Coding Standards

1. Code formatting:
 - Use a consistent code formatting style for readability and maintainability.
 - Follow PEP8 guidelines for Python code formatting.
2. Naming conventions:
 - Use clear and descriptive names for variables, functions, classes, and modules to enhance readability and minimize ambiguity.
 - Use camelCase for function and variable names, and PascalCase for class names.
3. Comments:
 - Include comments in the code to provide explanations for complex or unclear code sections.
 - Use descriptive comments to explain the purpose of the code, not just what it does.
4. Modularity:
 - Follow the principles of modularity to enhance code reuse and maintainability.
 - Separate code into logical modules and functions, each with a single responsibility.
5. Error handling:
 - Include error handling mechanisms in the code to handle expected and unexpected errors gracefully.
 - Provide clear error messages and logs to help with debugging and troubleshooting.
6. Testing:
 - Follow a testing framework such as unittest or pytest to ensure code quality and reliability.
 - Write unit tests for each function and module, and integration tests to test the system as a whole.
7. Security:
 - Ensure the code is secure by following best practices such as input validation, authentication, and authorization mechanisms.
 - Use secure protocols and libraries for handling sensitive data.

4.2. Screen Shots



4.2.1 login page

smit.bhathgara@gmail.com Applications Applicants AddJob Logout App-HomePage

Users

username	email	first_name	lastname	phone_number	staff_status	active_status	action
smit	smit@gmail.com	smit	bhathgara	9988776655	False	True	<button>Delete</button> <button>Edit</button>
admin	admin@gmail.com	admin	admin	9999999999	True	True	<button>Delete</button> <button>Edit</button>
admin2	admin2@gmail.com	admin2	admin2	9999999999	True	True	<button>Delete</button> <button>Edit</button>
asm	asm@gmail.com	asm	asm	9898706475	False	True	<button>Delete</button> <button>Edit</button>
smit0	smit.bhathgara@gmail.com	Smit	bhathgara	0992 410 9325	False	True	<button>Delete</button> <button>Edit</button>
admin3	admin3@gmail.com	smit	smiot	None	True	True	<button>Delete</button> <button>Edit</button>
helloadmin	helloadmin@gmail.com	hello	admin	9988776655	True	True	<button>Delete</button> <button>Edit</button>
raj@gmail.com	raj@gmail.com	raj	gj	9988776655	False	True	<button>Delete</button> <button>Edit</button>
lambodar	lambodaryuvakmandal@gmail.com	lambodar	yuvak mandal	9998887771	False	True	<button>Delete</button> <button>Edit</button>
smit_1907	smit.bhathgara@gmail.com	Smit	bhathgara	0992 410 9325	True	True	<button>Delete</button> <button>Edit</button>

4.2.2 Admin-side Manage Users

The screenshot shows a web application interface titled "Applications Page". At the top, there is a navigation bar with links: Applications, Applicants, AddJob, Logout, and App-HomePage. Below the title, there is a table with the following columns: Applicant, Applicant's email, Applicant's Job, CVs, Aproved, and Action. The table contains four rows of data:

Applicant	Applicant's email	Applicant's Job	CVs	Aproved	Action
smit bhathgara	smit@gmail.com	PythonDeveloper	CVs	False	Delete
smit bhathgara	smit@gmail.com	react	CVs	False	Delete
asm asm	asm@gmail.com	PythonDeveloper	CVs	False	Delete
smit smiot	admin3@gmail.com	PythonDeveloper	CVs	False	Delete

4.2.3 Admin-side Manage applications

The screenshot shows a web application interface titled "Applicants". At the top, there is a navigation bar with links: Applications, Applicants, AddJob, Logout, and App-HomePage. Below the title, there is a table with the following columns: username, email, first_name, lastname, Job, CVs, Aproved, and Action. The table contains four rows of data:

username	email	first_name	lastname	Job	CVs	Aproved	Action
smit	smit@gmail.com	smit	bhathgara	PythonDeveloper		False	Delete
smit	smit@gmail.com	smit	bhathgara	react		False	Delete
asm	asm@gmail.com	asm	asm	PythonDeveloper		False	Delete
admin3	admin3@gmail.com	smit	smit	PythonDeveloper		False	Delete

4.2.4 Admin-side Manage Applicants

smit.bathgara@gmail.com Applications Applicants AddJob Logout App-HomePage

Add Job

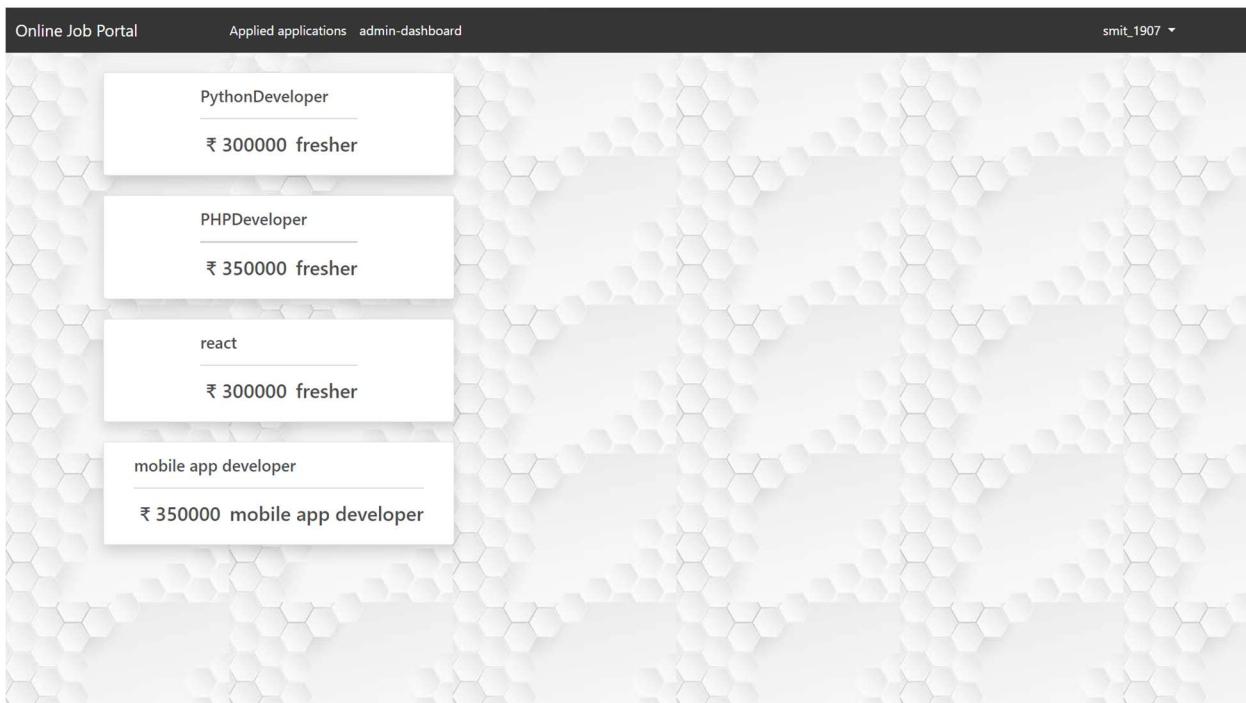
jobname	Add jobname
company	Add company name
description	description
pay	Add pay
address	Enter address
start Date	dd-mm-yyyy <input type="text"/>
End Date	dd-mm-yyyy <input type="text"/>

Jobs

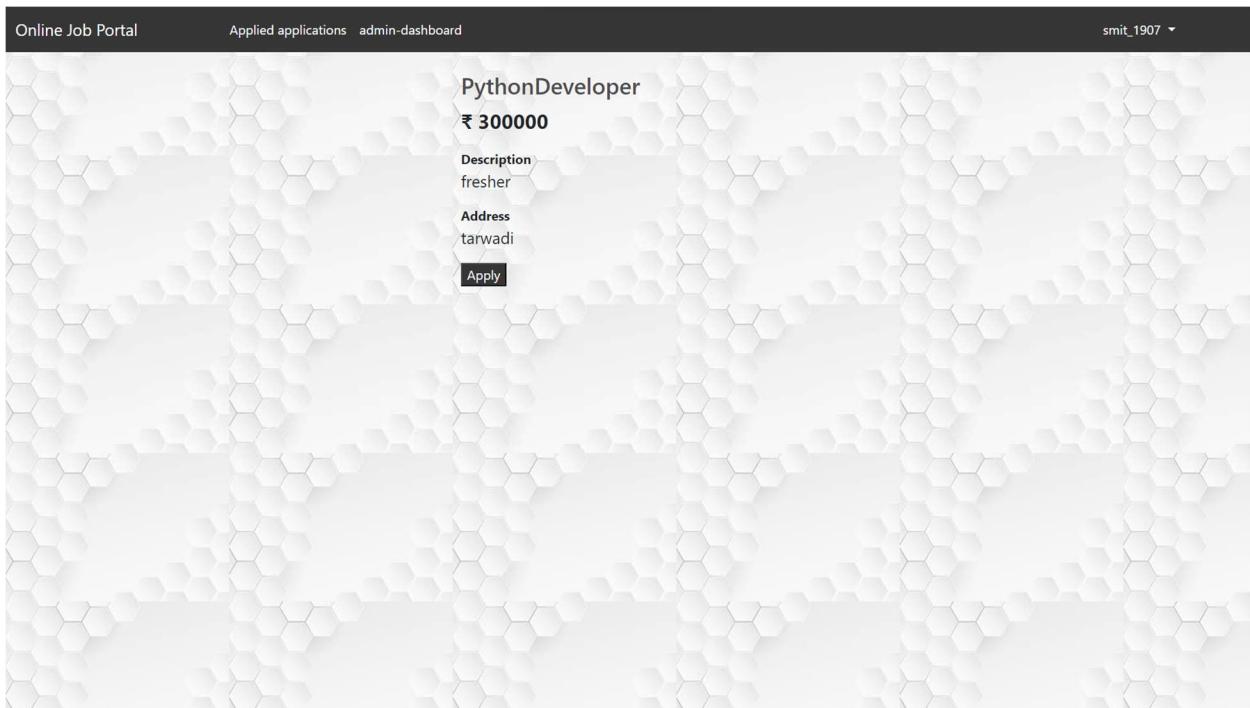
jobname	company	description	pay	address	application_start_date	application_end_date	action
PythonDeveloper	logixbuilt	fresher	300000	tarwadi	April 16, 2023	May 7, 2023	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
PHPDeveloper	alita	fresher	350000	adajan	April 25, 2023	May 7, 2023	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
react	logixbuilt	fresher	300000	tarwadi	Feb. 6, 2023	May 10, 2023	<input type="button" value="Delete"/> <input type="button" value="Edit"/>
mobile app developer	infosystem	mobile app developer	350000	adajan, surat	May 9, 2023	May 31, 2023	<input type="button" value="Delete"/> <input type="button" value="Edit"/>

127.0.0.1:8000/admin_side_EditJob/27/

4.2.5 Admin-side Manage Jobs



4.2.6 Home page



4.2.7 Job information

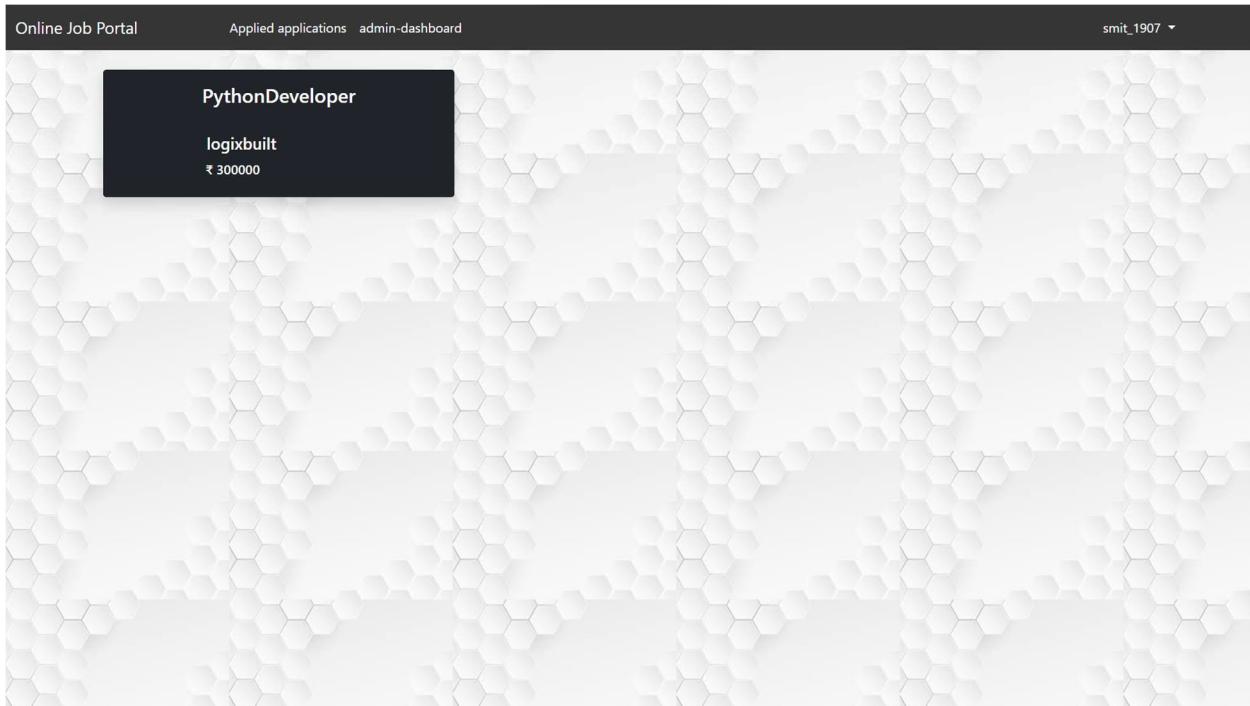
The screenshot shows the application form for the Python Developer position. The fields filled are:

- Name: Smit bhathgara
- Email: smit.bhathgara@gmail.com
- Phone: 9955446622
- Write about your self in brief:

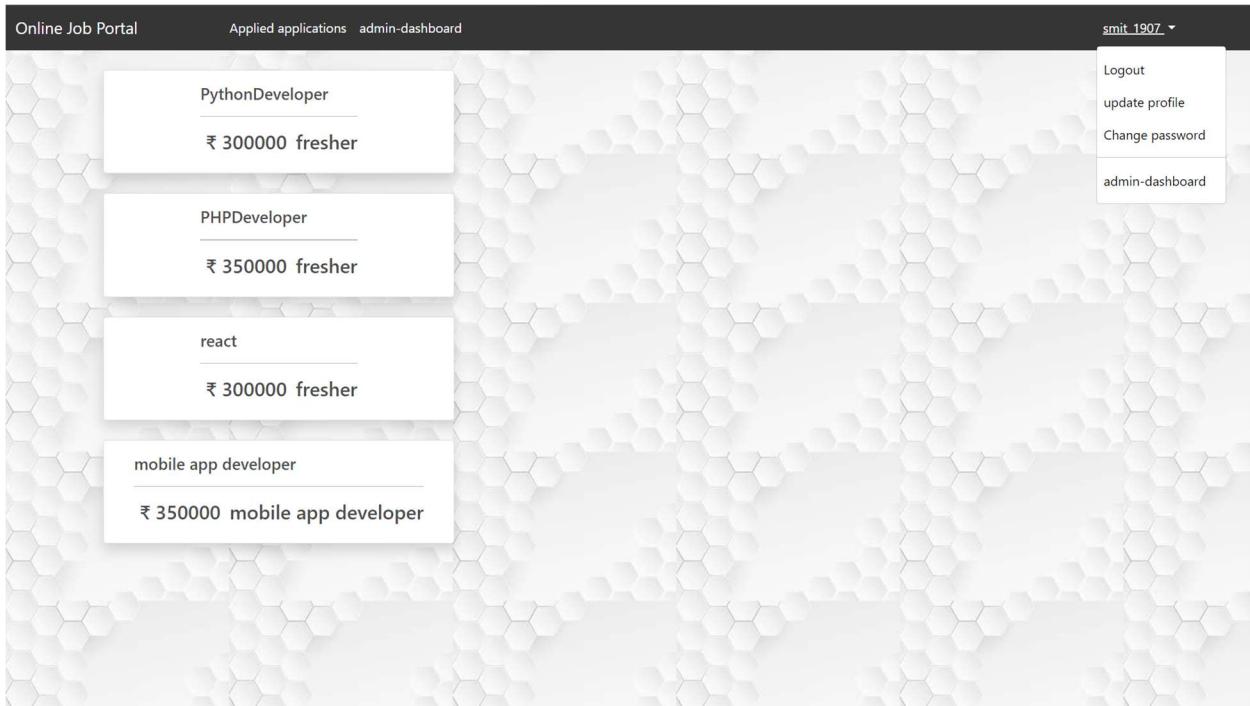
Hi, I'm smit bhathgara, a software developer specializing in Python. I've been passionate about technology and programming since a young age, and have pursued my interest in software development ever since. With 5 years of experience, I've worked on a variety of projects ranging from web development to machine learning and data analysis.
- Jobname: PythonDeveloper
- upload your CV: Choose File Python-Django_3.pdf

A "Submit" button is visible at the bottom left.

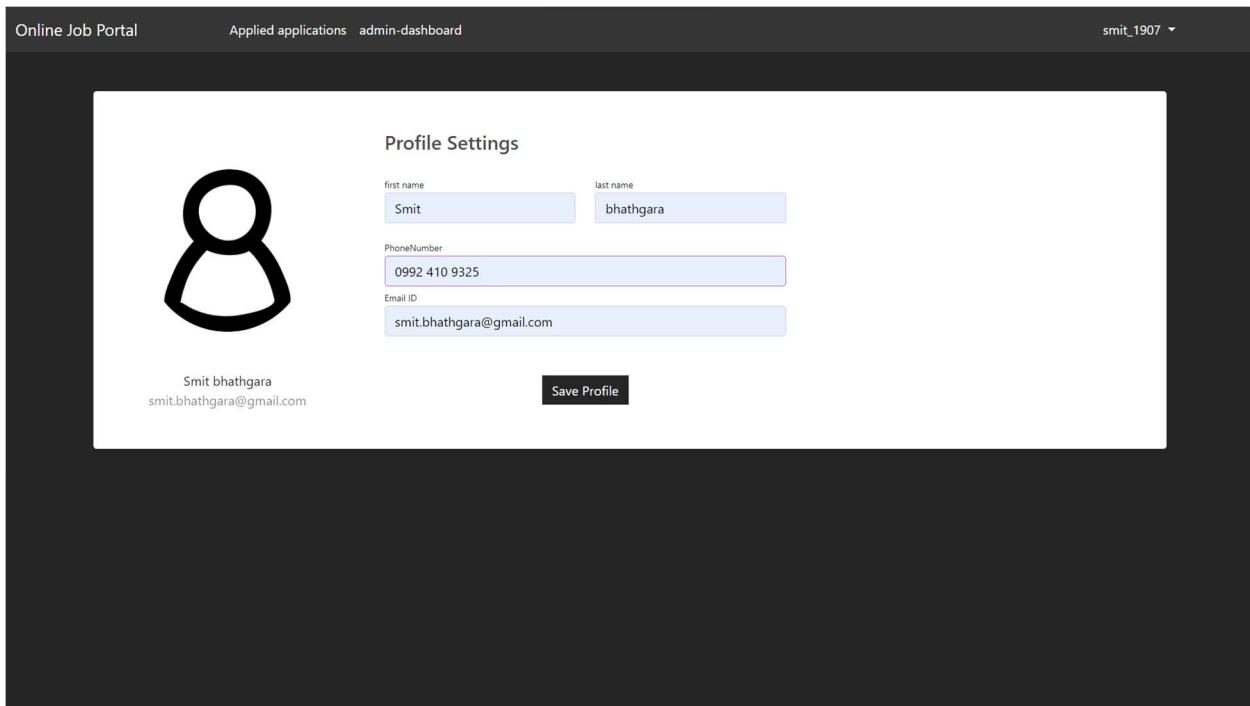
4.2.8 apply for job



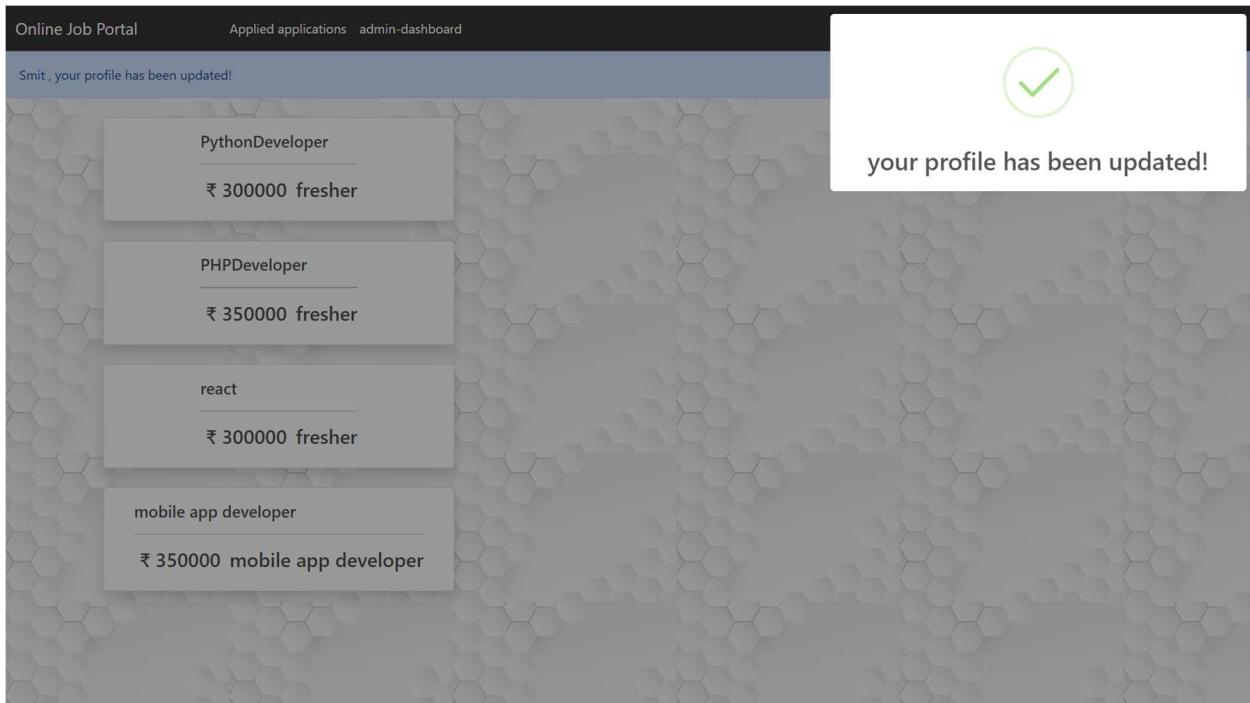
4.2.9 Applied jobs



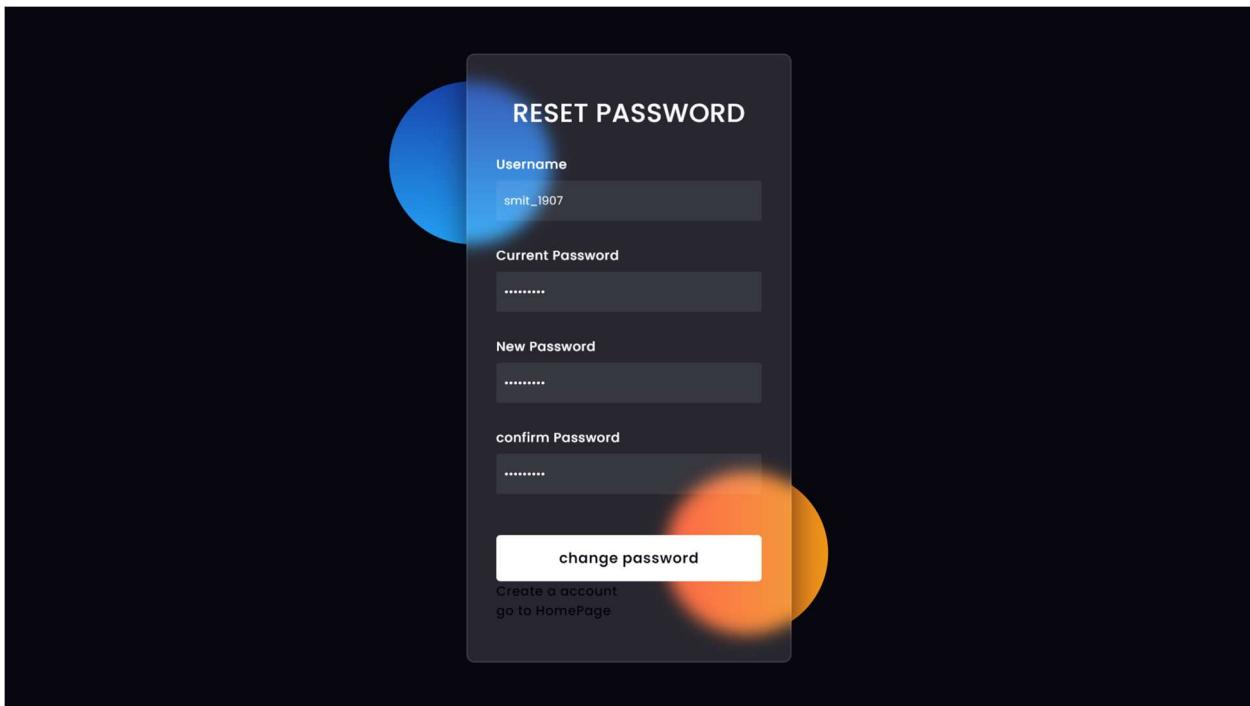
4.2.10 Home page options



4.2.11 User Profile



4.2.12 work success messages



4.2.13 Reset password

5

Agile Documentation

5.1 Agile Project Charter

General Project Details	
Project Name:	Question Paper Setter System
Project By:	Smit Bhathgara
Start Date:	01 / Feb / 2023
End Date:	03 / Mar / 2023
Project Details	
Exclusives Summary	The online job portal is a web-based platform connecting job seekers with potential employers, built using Django framework and SQLite database, with functionality for job search, application, and profile updating.
Objectives	The objective of the online job portal project is to provide a user-friendly and efficient platform for job seekers and recruiters to connect, allowing for easy job search, application, and management.

5.2 Agile Roadmap / Schedule

1st Quarter	2nd Quarter
1/Feb/2023 to 7/Feb/2023 Understand project definition, gather requirement, finalized the project scope, Target User, Core Component and to learn Laravel web Frame work and make an admin side and user side in Laravel project.	8/Feb/2023 to 14/Feb/2023 we design and set the system and draw various diagrams such as: <ul style="list-style-type: none">❖ Use case Diagram❖ Activity Diagram❖ Interaction Diagram❖ Class Diagram❖ System Flow For system and prepare data dictionary.
3rd Quarter	4th Quarter
15/Feb/2023 to 21/Feb/2023 I learn Django and prepared front-end side of our project and test validation in all pages and make document side by side.	22/Feb/2023 to 03/Mar /2023 Implement a proper coding structure for the system and implement Email authentications
Final Quarter	
6/May/2023 Final presentation and final documentation to be done.	

5.3 Agile Project Plan

Task Name	Responsible	Start	End	Days	Status
Sprint 1		1/Feb/2023	7/Feb/2023	7	Complete
Project Definition	Smit Bhathgara	1/Feb/23	1/Feb/23	1	Complete
Gathering requirement	Smit Bhathgara	1/Feb/23	1/Feb/23	1	Complete
Project Scope	Smit Bhathgara	1/Feb/23	1/Feb/23	1	Complete
Core Component	Smit Bhathgara	1/Feb/23	1/Feb/23	1	Complete
Targeted User	Smit Bhathgara	1/Feb/23	1/Feb/23	1	Complete
Learn Django	Smit Bhathgara	1/Feb/23	7/Feb/23	6	Complete
Sprint 2	Smit Bhathgara	8/Feb/2023	20/Feb/2023	13	Complete
Template set on front-end for user	Smit Bhathgara	8/Feb/23	11/Feb/23	4	Complete
Template set on front-end for admin	Smit Bhathgara	12/Feb/23	14/Feb/23	3	Complete
Managing users, jobs, and applications on the admin site	Smit Bhathgara	15/Feb/23	15/Feb/23	1	Complete
jobview and apply for jobs on the user site	Smit Bhathgara	16/Feb/23	17/Feb/23	3	Complete
Manage profile on the user site	Smit Bhathgara	16/Feb/23	17/Feb/23	3	Complete
Login – registration for user and admin	Smit Bhathgara	18/Feb/23	19/Feb/23	2	Complete
Google login	Smit Bhathgara	20/Feb/23	20/Feb/23	1	Complete
Sprint 3	Smit Bhathgara	21/Feb/2023	23/Feb/2023	3	Complete
Use-case diagram	Smit Bhathgara	21/Feb/23	21/Feb/23	1	Complete
Activity diagram	Smit Bhathgara	22/Feb/23	22/Feb/23	1	Complete
Class diagram	Smit Bhathgara	22/Feb/23	22/Feb/23	1	Complete
Interaction diagram	Smit Bhathgara	23/Feb/23	23/Feb/23	1	Complete
System flow	Smit Bhathgara	23/Feb/23	23/Feb/23	1	Complete
Sprint 4	Smit Bhathgara	23/Feb/2023	27/Feb/2023	6	Complete
UI design for Login page with validation	Smit Bhathgara	23/Feb/23	25/Feb/23	3	Complete
Apply Change Password with validation	Smit Bhathgara	26/Feb/23	27/Feb/23	3	Complete

Sprint 5	Smit Bhathgara	28/Feb/2023	06/May/2023	4	Complete
In Feature Applied change password	Smit Bhathgara	28/Feb/23	28/Feb/23	4	Complete
Documentation	Smit Bhathgara	1/Mar/2023	3/Mar/2023	1	Complete
Presentation	Smit Bhathgara	6/May/2023	6/May/2023	1	

5.4 Agile User Story (Minimum 3 Tasks)

- A user story is a tool used in agile software development to capture a description of a software feature from an end user perspective. A user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement for the system.
- A story point is a metric used in agile project management and development to estimate the difficulty of implementing a given user story.

Story Point	Story
1. Admin Logs	- As Admin, I can add, update, delete and view admin logs like users, jobs, applications and assign template data.
2. User Logs	- As user, I can add view jobs , apply for jobs , view applied jobs, update profile , building resume and login with google.
3. Validations	- Putting validations in entire Job Portal System

5.5 Agile Release Plan

Sprint	Task Name	Start	Finish	Duration	Status	Release Date
1	Project Definition	1/Feb/23	1/Feb/23	1	Released	6/Feb/23
1	Gathering requirement	1/Feb/23	1/Feb/23	3	Released	6/Feb/23
1	Project Scope	1/Feb/23	1/Feb/23	4	Released	9/Feb/23
1	Core Component	1/Feb/23	1/Feb/23	2	Released	13/Feb/23
1	Targeted User	1/Feb/23	1/Feb/23	3	Released	15/Feb/23
2	Learn Django	1/Feb/23	7/Feb/23	6	Released	20/Feb/23
2	Template set on front-end for user	8/Feb/23	11/Feb/23	1	Released	1/Mar/23
2	Template set on front-end for admin	12/Feb/23	14/Feb/23	2	Released	4/Mar/23
2	Managing users, jobs, and applications on the admin site	15/Feb/23	15/Feb/23	3	Released	7/Mar/23
2	jobview and apply for jobs on the user site	16/Feb/23	17/Feb/23	2	Released	11/Feb/23
2	Manage profile on the user site	16/Feb/23	17/Feb/23	3	Released	14/Mar/23
2	Login – registration for user and admin	18/Feb/23	19/Feb/23	2	Released	15/Mar/23
2	Google login	20/Feb/23	20/Feb/23	3	Released	16/Mar/23
3	Use-case diagram	21/Feb/23	21/Feb/23	4	Released	9/Apr/23
3	Activity diagram	22/Feb/23	22/Feb/23	2	Released	11/Apr/23
3	Class diagram	22/Feb/23	22/Feb/23	3	Released	12/Apr/23
3	Interaction diagram	23/Feb/23	23/Feb/23	4	Released	12/Apr/23
3	System flow	23/Feb/23	23/Feb/23	3	Released	13/Apr/23
4	UI design for Login page with validation	23/Feb/23	25/Feb/23	3	Released	15/Apr/23
4	Apply Change Password with validation	26/Feb/23	27/Feb/23	3	Released	16/Apr/23
5	In Feature Applied change the password	28/Feb/23	28/Feb/23	4	Released	17/Apr/23
5	Documentation	1/Mar/2023	3/Mar/2023	7	Released	20/Apr/23
5	Presentation	6/May/2023	6/May/2023	1	Released	6/May/23

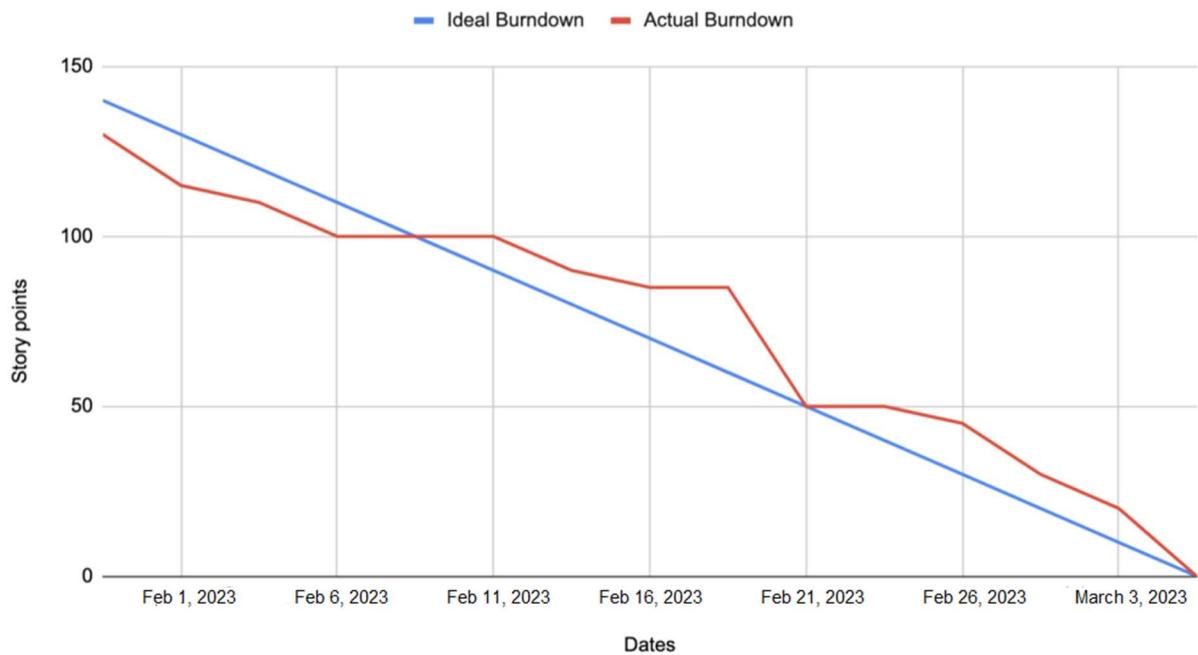
5.6 Agile Sprint Backlog

Backlog Task & Id	Assign To	Status	Original Estimate Days	Actual Results Days
User Story #1			13	11
Design login and registration page	Smit Bhathgara	Complete	3	3
Design home screen	Smit Bhathgara	Complete	2	3
Design admin side	Smit Bhathgara	Complete	4	2
Design user side	Smit Bhathgara	Complete	4	3
User Story #2			8	9
Login and registration screen	Smit Bhathgara	Complete	1	1
Home page	Smit Bhathgara	Complete	1	1
Profile update	Smit Bhathgara	Complete	1	1
Resume building	Smit Bhathgara	Complete	2	3
Change password	Smit Bhathgara	Complete	1	1
Apply for job	Smit Bhathgara	Complete	2	2
User Story #3			11	11
Admin side manage users	Smit Bhathgara	Complete	3	2
Admin side manage jobs	Smit Bhathgara	Complete	2	2
Admin side manage Applicants	Smit Bhathgara	Complete	2	3
Error massages and validations	Smit Bhathgara	Complete	3	2
logout	Smit Bhathgara	Complete	1	2

5.7 Agile Test Plan

Project Name		Liva-Contact Directory			
Written By		ET21MTCA004	Smit Bhathgara		
TESTED BY		ET21MTCA004	Smit Bhathgara		
Test	Date	Action	Expected Result	Actual Result	Pass
1	21-04-2023	Understand definition	Understand the objective of definition	Definition Understood	Pass
2	21-04-2023	Draw diagrams	All the diagrams are draw correctly	Complete Diagrams	Pass
3	22-04-2023	Crud Operation	Insert, Update and Delete should perform successfully	All crud operation is performed successfully	Pass
4	22-04-2023	Login page Invalid Details	You have entered valid login details	You have entered valid login details	Pass
5	22-04-2023	Login page valid Details	Redirect to Home Page	Redirect to Home Page	Pass
6	23-04-2023	Manage Profile	Profile updated successfully	Profile updated successfully	Pass
7	24-04-2023	To set home screen	To show all list of alljobs	To show all list of jobs	Pass
8	25-04-2023	To build resume	To build successfully resume	Resume build successfully	Pass
9	27-04-2023	Login with google	log in successfully with google account	Successfully logged in with google account	Pass
10	28-04-2023	Check validations	Checking all error messages and validations	All validations work successfully	Pass

5.8 Earned-value and burn charts



6. Proposed Enhancements

- Resume parsing for automatic extraction of relevant information from job seeker resumes.
- Automated job recommendations using machine learning algorithms based on job seeker profiles and job postings.
- Customized email notifications for job seekers and recruiters about new job postings, job applications, and status updates.
- Integration with social media platforms to allow job seekers to sign up and apply for jobs using their social media accounts.
- Integration with video interview platforms to allow recruiters to conduct video interviews directly through the job portal.
- Job posting analytics to track metrics such as the number of views, applications, and hires per job posting.

By implementing these proposed enhancements, the online job portal will become more user-friendly, efficient, and effective, making it a valuable tool for job seekers and recruiters alike.

7. Conclusion

- In conclusion, the development of an online job portal using Django framework and SQLite database offers numerous benefits for both job seekers and employers. The project aims to provide a user-friendly platform for job seekers to search and apply for jobs efficiently, and for employers to manage their job postings and find qualified candidates.
- Additionally, the project addresses limitations of the existing system and proposes enhancements such as job application tracking, employer dashboard and feedback system to improve the overall user experience. However, it is important to acknowledge the assumptions and constraints of the project, and consider the coding standards and security measures to ensure a reliable and secure job portal. Overall, the proposed enhancements and features of the online job portal project have the potential to greatly benefit the targeted users, job seekers, and employers, and enhance the effectiveness of the job portal in connecting job seekers with job opportunities.

8. Bibliography

- Django Documentation. Available at: <https://docs.djangoproject.com/>.
- SQLite Documentation. Available at: <https://sqlite.org/docs.html>.
- W3Schools. "HTML Tutorial." Available at: <https://www.w3schools.com/html/>.
- Python Software Foundation. "Python 3 Documentation." Available at: <https://docs.python.org/3/>.

Project 2

Web Scraping

1 Introduction

1.1. Existing System

- Currently, there is no dedicated system or software available that specifically fetches car parts data from eBay using automated web scraping techniques. Most users rely on manual searching and browsing through the eBay website to find and extract car parts data, which can be time-consuming and labour-intensive. Moreover, manual methods may not be efficient for handling a large volume of data or updating data in real time.
- Some existing systems or tools may provide general web scraping capabilities using programming languages or libraries, but they may not be tailored specifically for fetching car parts data from eBay. These tools may require significant effort and expertise to configure and customize for eBay's website structure and data format.
- Additionally, existing systems may not provide integration with databases such as PostgreSQL and MongoDB for efficient data storage and retrieval, or may not have the capability to handle data validation, error handling, and advanced data manipulation.
- As such, there is a need for a dedicated system that can effectively and efficiently fetch car parts data from eBay using automated web scraping techniques, and provide features such as data handling, storage, and manipulation using Python libraries such as Selenium, BeautifulSoup, Pandas, Psychopg2, and NumPy, and database integration with PostgreSQL and MongoDB. This project aims to fill this gap by developing a custom system that specifically caters to the needs of fetching car parts data.

1.2. Need for the New System

There are several reasons that highlight the need for a new system to fetch car parts data from eBay using automated web scraping techniques:

1. Time Efficiency: Manual searching and browsing through the eBay website to find and extract car parts data can be time-consuming and labour-intensive. Automating this process using a dedicated system can significantly reduce the time and effort required to fetch large volumes of data, resulting in increased efficiency.
2. Real-time Updates: Car parts data on eBay is dynamic and constantly changing. A dedicated system that can fetch data in real-time through automated web scraping allows for up-to-date and accurate data retrieval, ensuring that the latest information is available for analysis and decision-making.
3. Data Volume and Scalability: eBay is a large online marketplace with millions of car parts listings. Manually handling such a large volume of data can be challenging and error-prone. A dedicated system with automated web scraping capabilities can efficiently handle large volumes of data and scale to accommodate future growth in data volume.
4. Data Validation and Manipulation: Automated web scraping can provide data validation and manipulation capabilities, such as data cleaning, data enrichment, and data transformation, ensuring that the fetched data is accurate, consistent, and ready for analysis.
5. Database Integration: Integrating with databases such as PostgreSQL and MongoDB can provide efficient data storage, retrieval, and management capabilities. This allows for better organization, retrieval, and analysis of fetched data, enhancing the overall system's performance.
6. Customization: Existing systems or tools may not be tailored specifically for fetching car parts data from eBay. Developing a custom system allows for customization to suit the specific requirements of fetching car parts data, ensuring optimal performance and functionality.
7. Ease of Use: A user-friendly system that simplifies the process of fetching car parts data from eBay using Python with Selenium, BeautifulSoup, Pandas, Psychopg2, and NumPy, and integrating with PostgreSQL and MongoDB, can make the data retrieval process more accessible to users with varying levels of technical expertise.

In summary, the new system is needed to enhance the efficiency, accuracy, scalability, and data management capabilities of fetching car parts data from eBay, providing real-time updates, data validation, manipulation, and integration with databases, while offering ease of use and customization to cater to specific project requirements.

1.3. Objective of the New System

The main objective of the new system is to develop a robust and efficient system for fetching car parts data from eBay using automated web scraping techniques. The specific objectives of the new system are:

1. Automation: Develop a system that automates the process of fetching car parts data from eBay, utilizing Python with Selenium, Beautiful Soup, Pandas, Psychopg2, and NumPy. This will eliminate the need for manual searching and browsing, saving time and effort, and improving overall efficiency.
2. Real-time Updates: Ensure that the fetched data is up-to-date and accurate by retrieving car parts data from eBay in real-time. This will enable users to have access to the latest information for analysis and decision-making.
3. Scalability: Design a system that can handle large volumes of car parts data, accommodating the dynamic nature of eBay's marketplace. The system should be scalable to handle future growth in data volume and accommodate changes in the structure of the eBay website.
4. Data Validation and Manipulation: Implement data validation and manipulation techniques to ensure the accuracy and consistency of the fetched data. This may include data cleaning, data enrichment, and data transformation, to ensure that the data is reliable and ready for analysis.
5. Database Integration: Integrate with PostgreSQL and MongoDB databases to efficiently store, retrieve, and manage the fetched car parts data. This will enable users to organize and analyse the data effectively, and facilitate data retrieval and reporting.
6. Customization: Develop a system that can be customized to suit the specific requirements of fetching car parts data from eBay. This may include customization of search parameters, data retrieval frequency, and data storage formats, to meet the unique needs of the project.
7. User-friendly Interface: Design a user-friendly interface that simplifies the process of fetching car parts data from eBay. The system should be easy to use, with clear instructions and intuitive navigation, catering to users with varying levels of technical expertise.

In summary, the main objective of the new system is to automate the process of fetching car parts data from eBay using Python with Selenium, Beautiful Soup, Pandas, Psychopg2, and NumPy, and integrating with PostgreSQL and MongoDB databases. The system aims to provide real-time updates, ensure data accuracy, scalability, and customization, and offer a user-friendly interface for efficient and effective data retrieval and analysis.

1.4. Problem Definition

The existing system for fetching car parts data from eBay lacks automation and real-time updates, leading to manual and time-consuming data retrieval processes. Additionally, the lack of customization options, data validation, and integration with databases make it challenging to handle large volumes of car parts data efficiently. The need for a new system arises to address these limitations and provide a more streamlined and automated solution for fetching and managing car parts data from eBay. The key problems identified in the existing system are:

1. Manual and time-consuming process: The current system requires manual effort to fetch car parts data from eBay, resulting in time-consuming and inefficient data retrieval processes.
2. Lack of automation and real-time updates: The existing system does not automate the process of data retrieval and does not provide real-time updates, resulting in outdated and inaccurate data.
3. Limited customization options: The current system lacks customization options for search parameters, data retrieval frequency, and data storage formats, limiting its flexibility and adaptability to different requirements.
4. Data validation and manipulation challenges: The existing system does not implement robust data validation and manipulation techniques, leading to inconsistent and unreliable data.
5. Lack of integration with databases: The current system does not integrate with databases for efficient data storage and retrieval, resulting in data management challenges for large volumes of car parts data.

In summary, the problem definition for this project is to overcome the limitations of the existing system, including the lack of automation, real-time updates, customization options, data validation, and integration with databases, in order to develop a more efficient and scalable solution for fetching and managing car parts data from eBay.

1.5. Core Components

- Python - Selenium, BeautifulSoup, Pandas, Psychopog2, NumPy ,
- MongoDB, PostgreSQL

1.6. Project Profile

Project Title	Web Scraping
Organization	Logix Built InfoTech Surat.
Platform	Windows 10
Tech Stack	Python - Selenium, BeautifulSoup, Pandas, Psychopog2, NumPy
Database	MongoDB, PostgreSQL
Project Guide	Mr. Aashish Mishra
Submitted to	Sarvajanik College of Engineering & Technology
Presented by	Bhathgara Smit

1.7. Assumptions and Constraints

Assumptions:

- The eBay website structure and data format will remain consistent throughout the duration of the project.
- The project team will have access to stable and reliable internet connectivity for web scraping and data retrieval from eBay.
- The Python libraries (Selenium, BeautifulSoup, Pandas, Psychopg2, NumPy) used in the project will be available and compatible with the project's development environment.
- The PostgreSQL and MongoDB databases will be accessible and properly configured for data storage and retrieval.
- Sufficient hardware resources (CPU, RAM, storage) will be available for running the project code and databases.
- The project team has the necessary permissions and credentials to interact with the eBay website and databases for data retrieval and storage.

Constraints:

- The project is limited to fetching car parts data from eBay and does not include functionalities for purchasing, selling, or transaction-related operations.
- The project is dependent on the structure and data format of the eBay website, which may change over time, affecting the data retrieval process.
- The project's performance and data retrieval speed may be influenced by the internet speed, website responsiveness, and other external factors that are beyond the project's control.
- The project team may face challenges in handling large volumes of data, processing time-consuming operations, or implementing complex data validation and manipulation techniques within the project's scope.
- The project may have limitations in terms of scalability, extensibility, and adaptability to different e-commerce platforms other than eBay, as it is specifically designed for eBay data retrieval using the mentioned technologies.
- It's important to clearly state the assumptions and constraints of the project to set the context and limitations of the project's scope, and to manage expectations regarding the project's capabilities and limitations.

1.8. Advantages and Limitations of the Proposed System

Advantages:

- Efficient and automated data retrieval: The project utilizes web scraping techniques with Python and related technologies to efficiently retrieve car parts data from eBay, automating the data collection process and saving time and effort compared to manual data retrieval methods.
- Large-scale data collection: The project has the potential to collect a large volume of data from eBay, allowing for comprehensive and extensive analysis of car parts data for various types and categories.
- Data analysis and insights: The project's use of data processing and analysis tools such as Pandas, Numpy, and Psychopg2 enables in-depth analysis of the retrieved data, providing valuable insights for market research, inventory management, pricing analysis, and other applications.
- Customization and flexibility: The project can be customized and extended to cater to specific requirements, such as filtering data based on specific criteria, analyzing data from different eBay categories or regions, and integrating with other data sources or systems.
- Practical application: The project has real-world application potential, as it can provide valuable data for businesses, researchers, or automotive enthusiasts in the domain of car parts analysis, pricing trends, and market insights.

Limitations:

- Reliance on website structure: The project's web scraping technique relies on the current structure of eBay's website, and any changes in the website's structure may require updates or modifications to the scraping logic.
- Data accuracy and reliability: The project's results are dependent on the accuracy and reliability of the car parts data available on eBay, and any discrepancies or inconsistencies in the data may impact the project's findings.
- Legal and ethical considerations: The project needs to adhere to eBay's terms of use and policies, including limitations on web scraping and data usage, to avoid any legal or ethical issues.
- Technology limitations: The project's use of Python libraries, web scraping techniques, and other technologies may have limitations, such as version compatibility, technical challenges, or performance issues.
- Data availability: The availability and accessibility of car parts data on eBay may vary, and some specific data points or categories may not be available or may require additional permissions or access.

2

Requirement Determination & Analysis

2.1. Requirement Determination

- User requirements: retrieving data the car parts from eBay as per the specific needs and expectations of the users who will be utilizing the car parts data retrieved from eBay. In users may include businesses, researchers, or automotive enthusiasts who require data for market analysis, pricing trends, inventory management, or other purposes.
- Data requirements: the specific data points or categories of car parts that need to be retrieved from eBay, such as part name, part number, manufacturer, price, location, condition, and other relevant attributes.
- Functional requirements: functionalities and features that the project should possess to meet the project objectives, such as web scraping capabilities using Selenium and Beautiful Soup, data processing and analysis using Pandas and NumPy, database integration using Psychopg2.
- Performance requirements: Define the performance benchmarks or criteria that the project needs to meet, such as data retrieval speed, data accuracy, data processing efficiency, and system responsiveness.
- Technology requirements: Python, Selenium, Beautiful Soup, Pandas, NumPy, Psychopg2, MongoDB, PostgreSQL.

2.2. Targeted Users

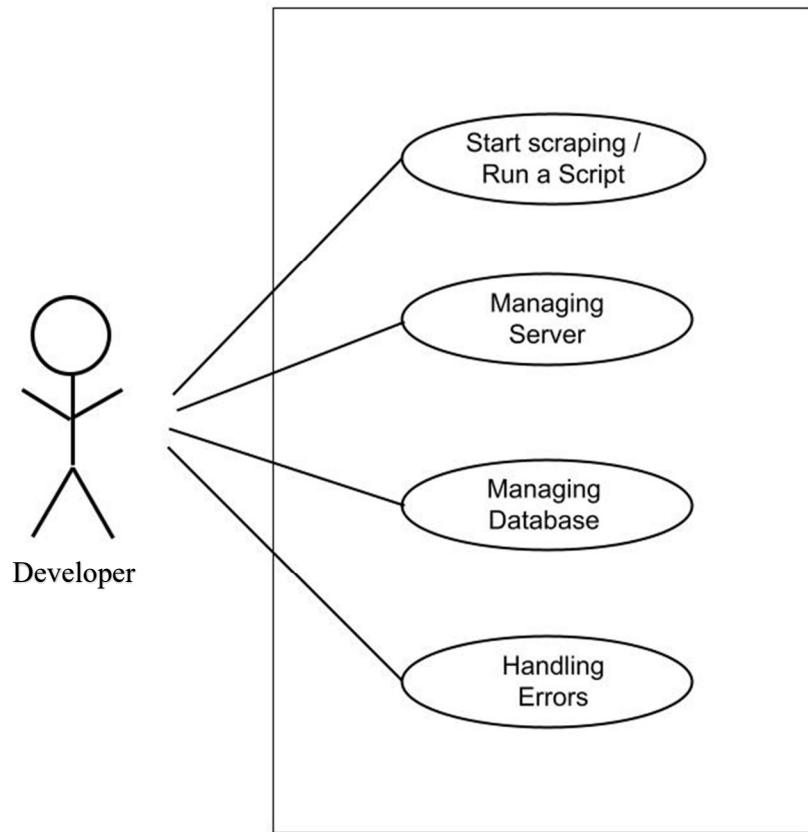
As our client is Business Analyst and Researcher, this script will be used for his analysis work. But this script and the data outcome is also may use for as following,

- Automotive Businesses: Car repair shops, auto parts stores, and dealerships.
- Automotive Enthusiasts: Car enthusiasts who restore, modify, or customize cars.
- Market Researchers: Researchers conducting market research in the automotive industry.
- Researchers and Academics: Researchers and academics studying the automotive field.
- Automotive Professionals: Mechanics, engineers, and technicians in the automotive industry.
- General Consumers: Consumers interested in purchasing car parts for their personal vehicles.

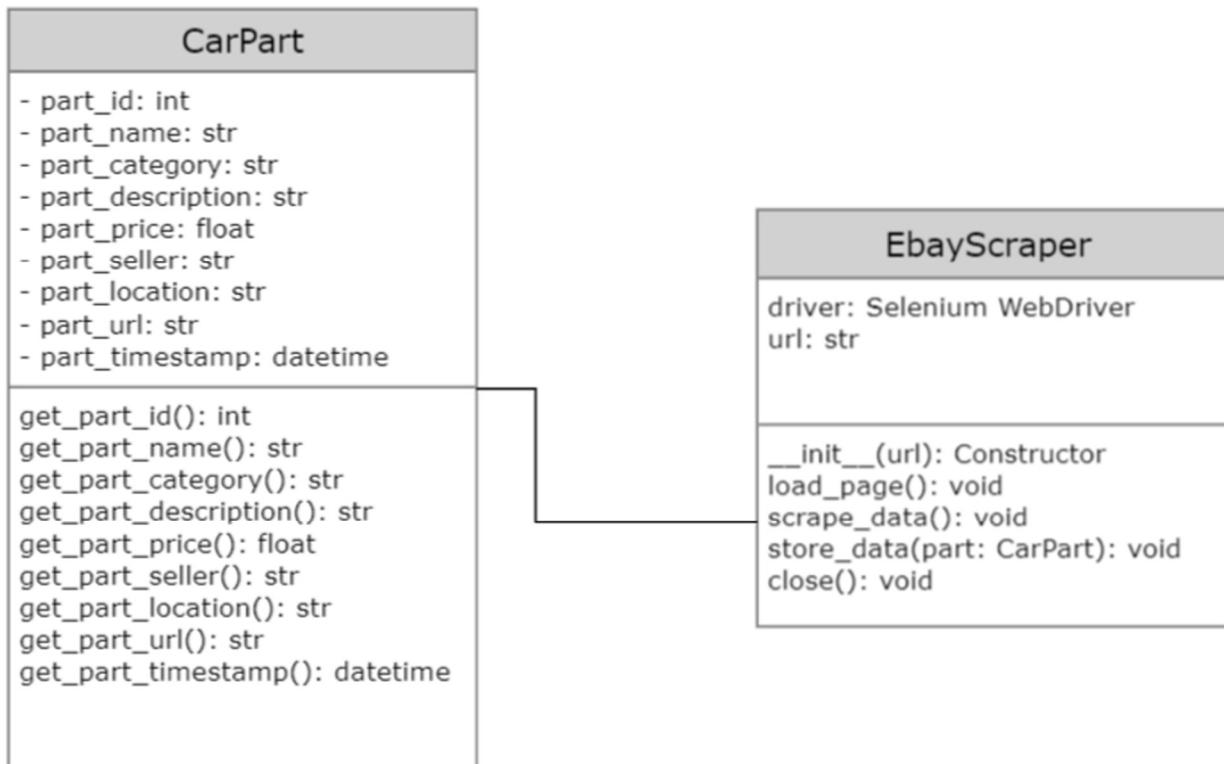
3

System Design

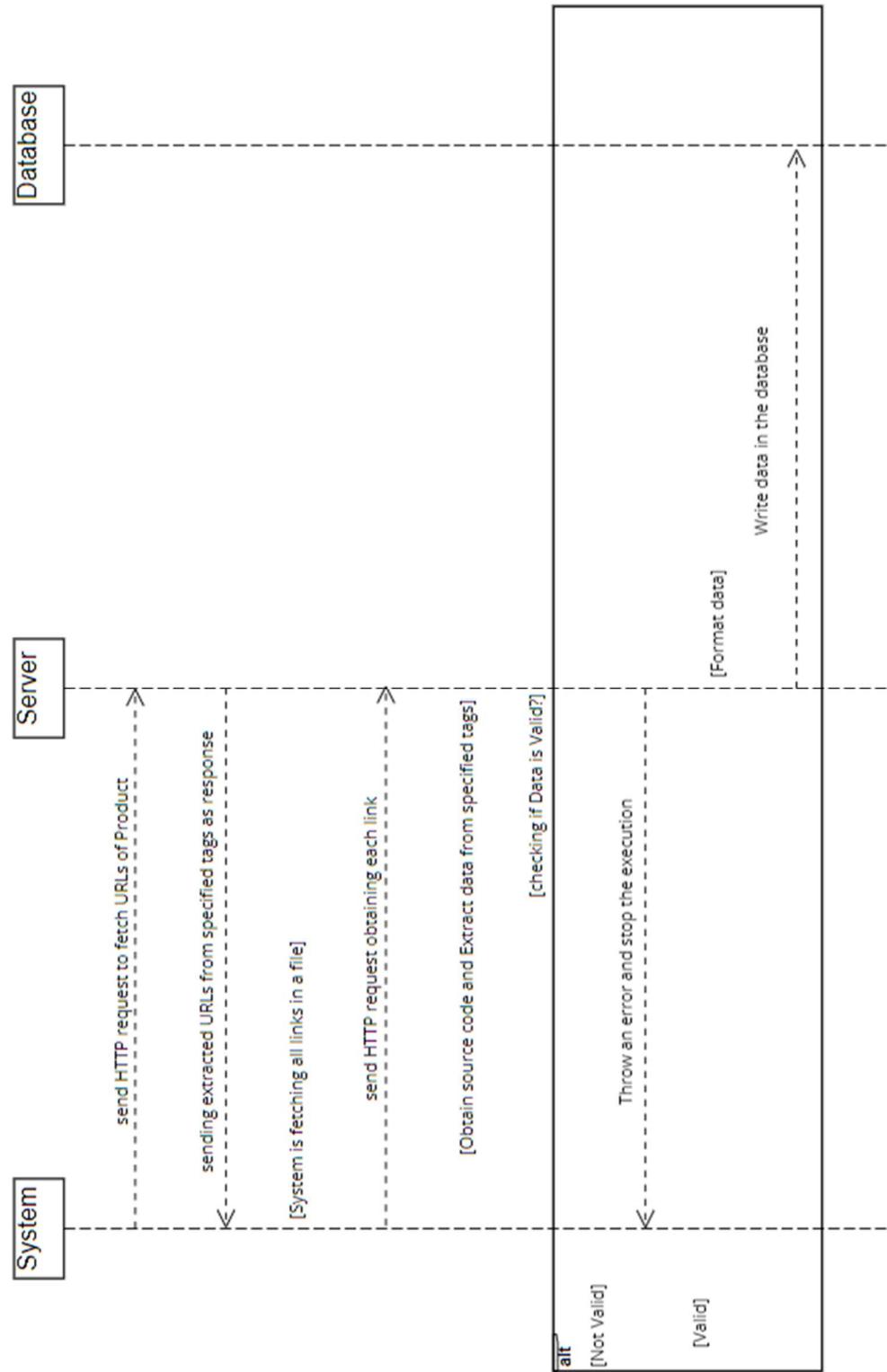
3.1. Use Case Diagram



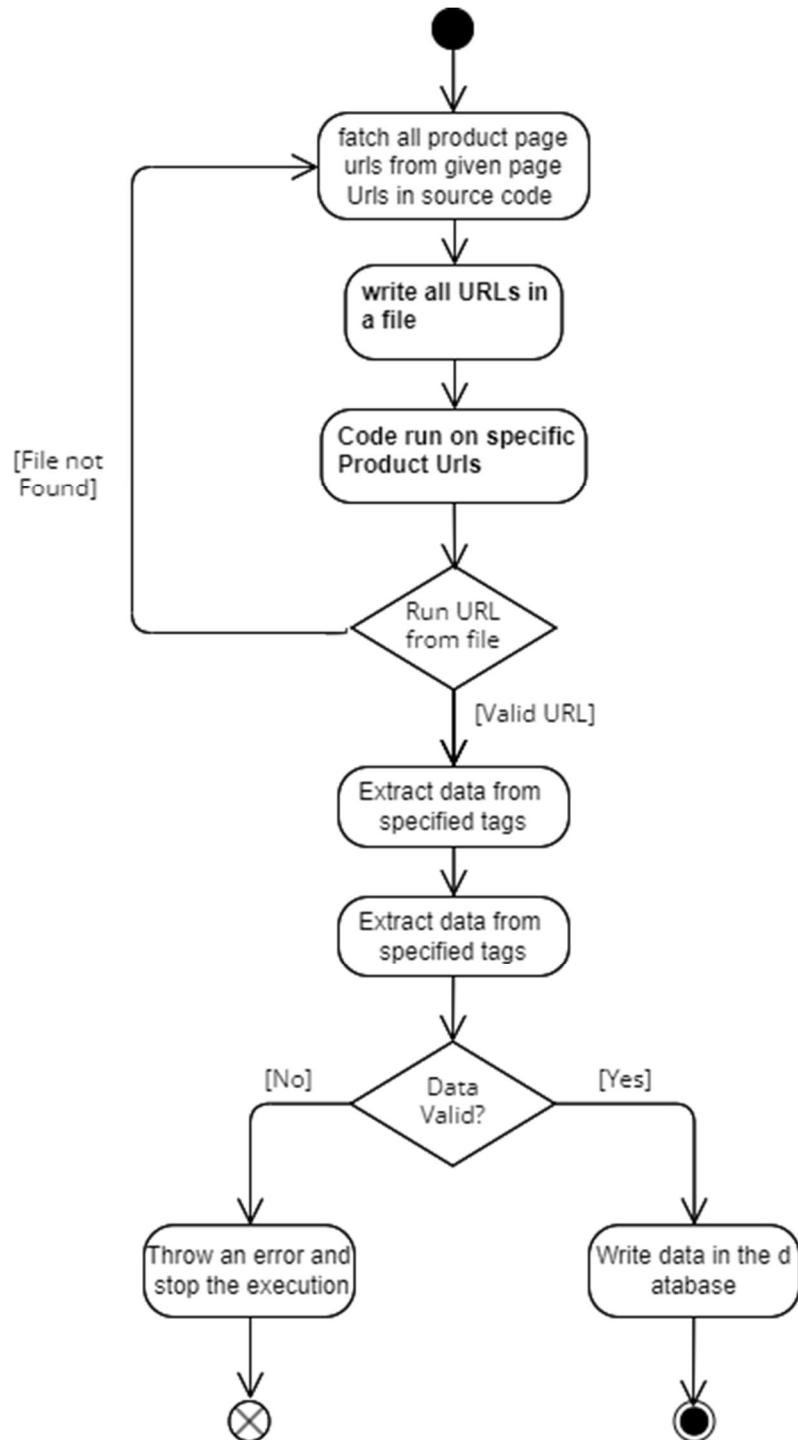
3.2. Class Diagram



3.3. Interaction Diagram



3.4. Activity Diagram



4

Development

4.1. Coding Standards

This Project follow all the coding-standards of Python such as Consistent Naming Conventions, Indentation and Formatting, Modularity and Reusability, Comments and Documentation, Error Handling and Exception Handling, Code Review and Testing , Security Considerations, Version Control etc.

1. Consistent Naming Conventions:

- Variables: Use camelCase or snake_case for variable names, e.g. carModel, part_price.
- Functions: Use lowercase with words separated by underscores for function names, e.g. fetch_data(), process_data().
- Classes: Use PascalCase for class names, e.g. CarPart, DataFetcher.
- Modules: Use lowercase with words separated by underscores for module names, e.g. ebay_data.py, car_parts.py.

2. Indentation and Formatting:

- spaces or tabs for indentation, and align code blocks consistently.
- Used proper spacing between operators, parentheses, and commas to improve code clarity.

3. Modularity and Reusability:

- Break down a complex data-fetching task into smaller functions or classes, such as fetch_data_from_ebay(), parse_data(), and store_data().
- Used functions and classes with clear and specific purposes, so that they can be easily reused in other parts of the project or in future projects.

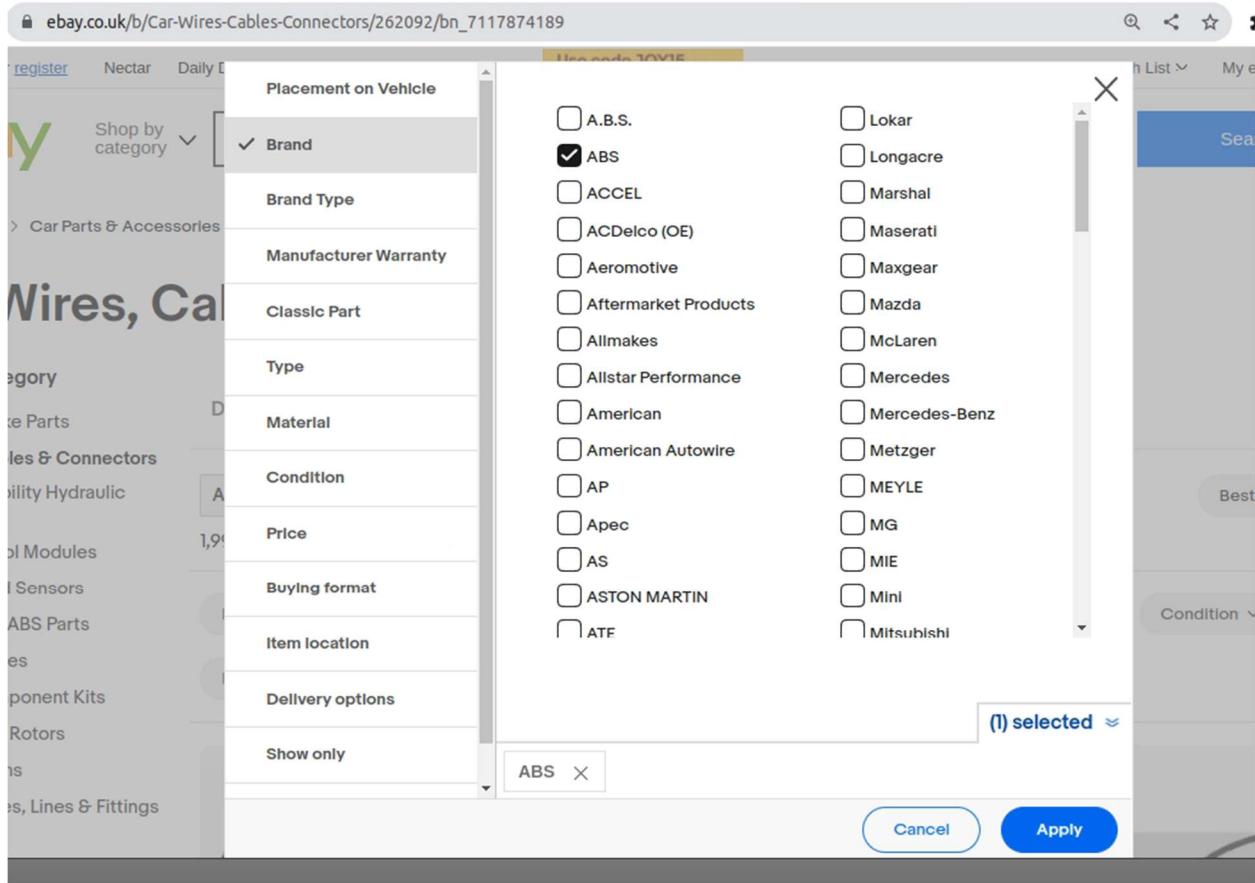
4. Comments and Documentation:

- Added comments to explain complex or tricky parts of the code, and provided inline comments for code that may not be immediately obvious.
- Include documentation's link that describes the purpose, inputs, and outputs of functions or classes, to help other developers understand and use code effectively.

5. Error Handling and Exception Handling:

- Use try-except blocks to catch and handle specific exceptions, and provide meaningful error messages to assist with debugging and troubleshooting.
- Security Considerations: Follow best practices for coding securely to protect against potential vulnerabilities and attacks. For example:
- Sanitize user inputs to prevent cross-site scripting (XSS) and cross-site request forgery (CSRF) attacks.

4.2. Screen Shots

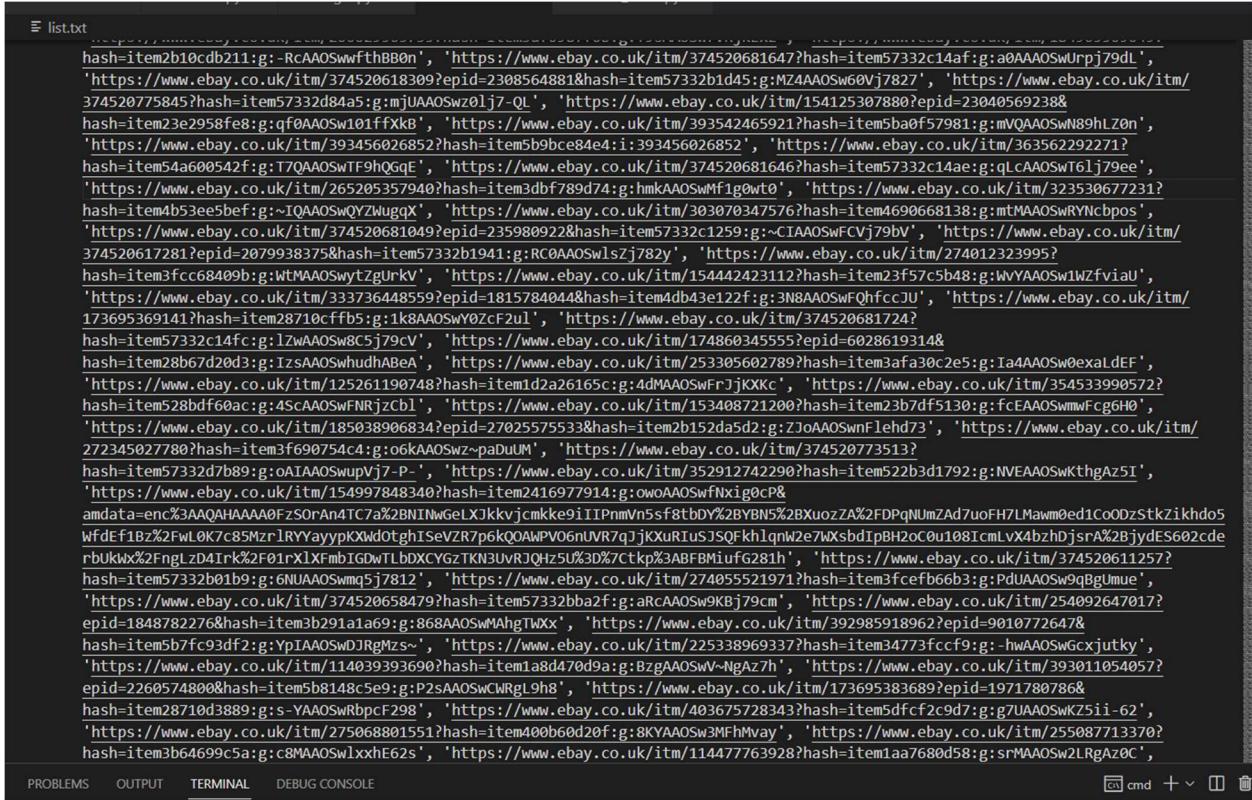


4.2.1 selecting category and coping URL of the page

```
44  MDX_Images - 0
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   ROBOT DOCUMENTATION   ROBOT OUTPUT

on page - 2 @ Brand Moroso
on page - 1 @ Brand Mr Gasket
on page - 1 @ Brand MSD Ignition
on page - 1 @ Brand Niles
on page - 1 @ Brand Nissan
on page - 1 @ Brand NK
on page - 2 @ Brand NK
on page - 3 @ Brand NK
on page - 1 @ Brand NPS
on page - 1 @ Brand OEM
on page - 2 @ Brand OEM
on page - 1 @ Brand OJD (Quick Brake)
on page - 1 @ Brand Opel
on page - 1 @ Brand Pagid
on page - 1 @ Brand Painless Wiring
on page - 2 @ Brand Painless Wiring
on page - 1 @ Brand Perma-Cool
on page - 1 @ Brand Petrol
on page - 1 @ Brand Peugeot
on page - 1 @ Brand Peugeot
on page - 1 @ Brand Pinnacle
on page - 1 @ Brand Pioneer
on page - 1 @ Brand Porsche
on page - 1 @ Brand PSA Peugeot Citroen
on page - 1 @ Brand Quickcar Racing Products
on page - 2 @ Brand Quickcar Racing Products
on page - 1 @ Brand Renault
on page - 1 @ Brand Renault
on page - 1 @ Brand Rover
on page - 1 @ Brand RPC
on page - 1 @ Brand Sealey
on page - 1 @ Brand Seat
on page - 1 @ Brand Standard Motor Products
```

4.2.2 Printing a Product name



```
list.txt
hash-item2b10cdb211:g:-RcAAOSwrfthB0n', 'https://www.ebay.co.uk/itm/374520681647?hash=item57332c14af:g:a0AA0SwUrpj79dI',
'https://www.ebay.co.uk/itm/374520618309?epid=2308564881&hash=item57332b1d45:g:Mz4AAOSw60Vj7827', 'https://www.ebay.co.uk/itm/374520775845?hash=item57332d84a5:g:mjUAAOSwz0lj7-QL', 'https://www.ebay.co.uk/itm/154125307880?epid=23040569238&
hash-item23e2958fe8:g:qf0AAOSw10ffXkB', 'https://www.ebay.co.uk/itm/393542465921?hash=item5ba0f57981:g:mQAAOSwN89hLZ0n',
'https://www.ebay.co.uk/itm/393456926852?hash=item5b9bce84e4:i:393456926852', 'https://www.ebay.co.uk/itm/363562292271?
hash-item54a00542f:g:T7QAAOSwTf9hGgE', 'https://www.ebay.co.uk/itm/374520681646?hash=item57332c14ae:g:qlcAAOSwT6lJ79ee',
'https://www.ebay.co.uk/itm/265205357940?hash=item3dbf789d74:g:hmkAAOSwMfg0wt0', 'https://www.ebay.co.uk/itm/323530677231?
hash-item4b53ee5bef:g:~tQAAOSwQYZLwuggX', 'https://www.ebay.co.uk/itm/303070347576?hash=item4690668138:g:mMAAOswRYNcbpos',
'https://www.ebay.co.uk/itm/374520681049?epid=235980922&hash=item57332c1259:g:~CIAOSwFCVj79bv', 'https://www.ebay.co.uk/itm/374520617281?epid=207938375&hash=item57332b1941:g:RC0AAOSwlszj782y', 'https://www.ebay.co.uk/itm/274012323995?
hash-item3fc68409b:g:WtMAAOswytzgurKV', 'https://www.ebay.co.uk/itm/154442423112?hash=item23f57c5b48:g:WVYAAOSw1WZfviaU',
'https://www.ebay.co.uk/itm/33736448559?epid=1815784044&hash=item4db43e122f:g:3N8AAOSwFQhfccJU', 'https://www.ebay.co.uk/itm/173695369141?hash=item28710cffb5:g:1k2AAOSwY0ZcF2ul',
'https://www.ebay.co.uk/itm/174860345559?epid=6028619314&
hash-item5332c14fc:g:IzwAAOSw8C5j79CV', 'https://www.ebay.co.uk/itm/174860345559?epid=6028619314&
hash-item28b67d20d3:g:IzsAAOSwhudhABeA', 'https://www.ebay.co.uk/itm/253305602789?hash=item3afa30c2e5:g:Ia4AAOSw0exaLdEF',
'https://www.ebay.co.uk/itm/125261190748?hash=item2a26165c:g:4dMAAOswFrJjkXKC', 'https://www.ebay.co.uk/itm/354533990572?
hash-item528bdf60ac:g:45cAAOSwFNrjzCbl', 'https://www.ebay.co.uk/itm/153408721200?hash=item23b7df5130:g:fcEAAOSwmwFcG6H0',
'https://www.ebay.co.uk/itm/185038906834?epid=27025575533&hash=item2b152da5d2:g:Zj0AAOSwnFlehd73', 'https://www.ebay.co.uk/itm/272345027780?hash=item3f690754c4:g:o6kAAOSwz-paDuUM', 'https://www.ebay.co.uk/itm/374520773513?
hash-item57332d7b89:g:oIAAAOSwupVj7-P-', 'https://www.ebay.co.uk/itm/352912742290?hash=item522b3d1792:g:NVEAAOSwKthgAz5I',
'https://www.ebay.co.uk/itm/154997848340?hash=item2416977914:g:ow0AAOSwfNxig0cP&
amdata=enc%3AAQAHAAA0FzSOrAn4TC7a%2BNINwGeLXJkkvjmke9iIPmVn5sf8tbdY%2BYBN5%2BXuoZAZ%2FDPqNUmzAd7uoFH7LMawm0ed1CoDzStkZikhdo5
WfdEf1Bz%2FwL0K7c85Mzr1rYYayppKXwd0tghISevZR7p6kQOAWPV06nUV7qjJkXuRTuS1SQFkhlnW2e7WxsbdiPbH2oC0u108IcmLvxb4bzhdjsraX2BjydeS602cde
rbUKwx%2FngLzD41rk%2F0r1XxFmbIGdwTlbDXCYGzTKN3uVnJQH5u%3D%7Ctkp%3ABFBMiufG281h', 'https://www.ebay.co.uk/itm/374520611257?
hash-item57332b01b9:g:6NUAAOSwmq5j7812', 'https://www.ebay.co.uk/itm/274055521971?hash=item3fcefb66b3:g:PduAAOSw9qBgUmue',
'https://www.ebay.co.uk/itm/374520658479?hash=item57332bba2f:g:arCAAOsw9Kbj79cm', 'https://www.ebay.co.uk/itm/254092647017?
epid=1848782276&hash-item3b291a1a69:g:868AAOSwMahgtKxx', 'https://www.ebay.co.uk/itm/392985918962?epid=9010772647&
hash-item5b7fc93df2:g:YpIAAOswDJrgMzs~, 'https://www.ebay.co.uk/itm/225338969337?hash-item34773fccf9:g:-hwAAOSwGcxjutky',
'https://www.ebay.co.uk/itm/114039393690?hash=itemd470d9a:g:BzgAAOSwv-NgA7h', 'https://www.ebay.co.uk/itm/393011054057?
epid=2260574800&hash-item5b8148c5e9:g:P2sAAOSwCMrgl9h8', 'https://www.ebay.co.uk/itm/173695383689?epid=1971780786&
hash-item28710d3889:g:s-YAAOSwRpccf298', 'https://www.ebay.co.uk/itm/403675728343?hash=item5dfcf2c9d7:g:g7UAAOSwKZ5ii-62',
'https://www.ebay.co.uk/itm/275068801551?hash=item400b60d20f:g:8KYAAOSw3MFhVmav', 'https://www.ebay.co.uk/itm/255087713370?
hash-item3b64699c5a:g:c8MAAOswLxxhE62s', 'https://www.ebay.co.uk/itm/114477763928?hash=item1aa7680d58:g:srMAAOsw2LRgAz0C',
```

4.2.3 creating a list of URLs for all Pages

The screenshot shows the PyCharm IDE interface with the Variables tool window open. The Variables window displays a hierarchical tree of variable values from a MongoDB query result. At the top level, there are variables like `children`, `collection_name`, `count_2`, `data` (containing 'BTB657'), `dbname`, and `dic`. Below `dic` is a URL entry. Under the URL, the eBay item details are listed: 'eBay Item Number', 'Title' (Triumph GT6 TR3A TR4 TR5 TR6 Brass Brake Pipe Union 3 Way Connector BTB657), 'Postage Cost', 'location' (located in London, United Kingdom), 'Condition' (None), 'Seller ID' (usefulcarpartsuk), 'Price' (£10.80), and 'images' (two URLs). Further down, there are fields for 'Fitment', 'Brand' (Triumph), 'Number', 'UPC', 'ISBN', 'Type' (Brake Pipe Fittings), 'EAN', 'Manufacturer Number', and 'Model'. The bottom section of the Variables window contains sections for 'WATCH', 'CALL STACK', and 'BREAKPOINTS'. The 'BREAKPOINTS' section has checkboxes for 'Raised Exceptions' (unchecked), 'Uncaught Exceptions' (checked), 'User Uncaught Exceptions' (unchecked), and two entries for 'single.py package' (both checked).

```
> children: <div class="ux-labels-values__values-content"><div><span class="ux-textspans ux-textspans--BOLD">FREE</span><span class="ux-textspans"> Eco-  
> collection_name: Collection(Database(MongoClient(host=['ec2-3-218-67-140.compute-1.amazonaws.com:27017']), document_class=dict, tz_aware=False, connect=True), 'Item_Specifics')  
> count_2: 13  
> data: 'BTB657'  
> dbname: Database(MongoClient(host=['ec2-3-218-67-140.compute-1.amazonaws.com:27017']), document_class=dict, tz_aware=False, connect=True), 'Item_Specifics')  
> dic: {'url': 'https://www.ebay.co...Swl89ft6wA', 'eBay Item Number': '225089870630', 'Title': 'Triumph GT6 TR3A TR4..57 | eBay', 'Postage Cost': 'FREE\x00conomy Delivery ', 'locat...  
> special variables  
> function variables  
'url': 'https://www.ebay.co.uk/itm/225089870630?hash=item346866db26:g:l8cAA0Swl89ft6wA'  
'eBay Item Number': '225089870630'  
'Title': 'Triumph GT6 TR3A TR4 TR5 TR6 Brass Brake Pipe Union 3 Way Connector BTB657 | eBay'  
'Postage Cost': 'FREE\x00conomy Delivery '  
'location': 'located in: london, united kingdom'  
'Condition': None  
'Seller ID': 'usefulcarpartsuk'  
'Price': '£10.80'  
> 'images': ['https://i.ebayimg.co...-l1000.jpg', 'https://i.ebayimg.co...-l1000.jpg']  
'Fitment': None  
'Brand': 'Triumph'  
'Number': 'BTB657'  
'UPC': ''  
'ISBN': ''  
'Type': 'Brake Pipe Fittings'  
'EAN': ''  
'Manufacturer Number': 'BTB657'  
'Model': ''  
> WATCH  
> CALL STACK  
> BREAKPOINTS  
  ■ Raised Exceptions  
  ☐ Uncaught Exceptions  
  ■ User Uncaught Exceptions  
  ● single.py package  
  ● single.py package
```

4.2.4 Debugging window to see what data insert into an object

4.2.5 Debugging window to see what data insert into an object

Item_Specifics.popsicle

1.8k DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION

1 - 20 of 1808

```
_id: ObjectId('63f855993f0223d5f886a8cb')
Condition: "New: A brand-new, unused, unopened and undamaged item in original reta..."
Type: "Brake Pipe Fittings"
Manufacturer Part Number: "BTB657"
Brand: "Triumph"
Ebay Item Number: "225089870630"

_id: ObjectId('63f855a177a735ff3586a8cb')
Condition: "New"
Warranty: "1 Year"
Brand: "Open Wheel & Sprint Cars Auto Parts"
Manufacturer Part Number: "ALL99145"
Alternate Description: "Rep Bulb and Socket for Allstar Gauges"
Brand Name: "ALLSTAR PERFORMANCE"
Package Length (in): "16.5"
Package Width (in): "16.5"
Part Number: "ALL99145"
Part Type: "#REF!"
Quantity: "Sold individually"
```

Sold individually

4.2.6 non-static field Data fetching in to MongoDB Database

eu-west-2.console.aws.amazon.com/ec2/home?region=eu-west-2#Instances:

New EC2 Experience Tell us what you think

Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
Scrap_	I-013b2abb92dab0d5	Stopped	t3.2xlarge	-	No alarms	eu-west-2
free tier instance	I-037fb9e0fbfadbb1b	Running	t2.micro	⚠️ 1/2 checks passed	No alarms	eu-west-2

Select an instance

4.2.7 AWS Server

G	H	I	J	K
Category ID	Scrape	Category URL	Total No of Listings	Status
262060	1	https://www.ebay.co.uk/b/262060	235199	
262061	1	https://www.ebay.co.uk/b/262061	91165	
262062	1	https://www.ebay.co.uk/b/262062	68103	
262063	1	https://www.ebay.co.uk/b/262063	109924	
262066	1	https://www.ebay.co.uk/b/262066	134	
262065	1	https://www.ebay.co.uk/b/262065	382	
38728	1	https://www.ebay.co.uk/b/38728	49085	
262067	1	https://www.ebay.co.uk/b/262067	3323	
OP	184616	1 https://www.ebay.co.uk/b/184616	12199	
174593	1	https://www.ebay.co.uk/b/174593	81092	
61523	1	https://www.ebay.co.uk/b/61523	128689	
258038	1	https://www.ebay.co.uk/b/258038	6158	
262068	1	https://www.ebay.co.uk/b/262068	1342	
262069	1	https://www.ebay.co.uk/b/262069	145	
33659	1	https://www.ebay.co.uk/b/33659	763792	
43946	1	https://www.ebay.co.uk/b/43946	52194	
33557	1	https://www.ebay.co.uk/b/33557	589716	
33550	1	https://www.ebay.co.uk/b/33550	2748	
33551	1	https://www.ebay.co.uk/b/33551	29391	
33660	1	https://www.ebay.co.uk/b/33660	435711	
33554	1	https://www.ebay.co.uk/b/33554	366951	
S	262071	1 https://www.ebay.co.uk/b/262071	29096	
Pa	33553	1 https://www.ebay.co.uk/b/33553	34264	
33555	1	https://www.ebay.co.uk/b/33555	298579	
262072	1	https://www.ebay.co.uk/b/262072	46031	
22556	1	https://www.ebay.co.uk/b/22556	45827	

PandA(Sep2022)

4.2.1 static field Data fetching in to PostgreSQL Database

5. Proposed Enhancements

- In this project we are not currently provide any UI for user to run the application , which is quite boring and confusing for non-technical clients , so it will be not problem after updating this project.
- Fetching Data from website is taking too much time, reduce time for fetching data.
- Make a better data accuracy, data processing efficiency, and system responsiveness

6. Conclusion

- In conclusion, this project has successfully demonstrated the capability to fetch data for car parts from eBay using web scraping techniques with Python. The use of Selenium and Beautiful Soup libraries enabled the extraction of data from web pages, while Pandas, Psychopg2, and Numpy provided efficient data handling, storage, and manipulation capabilities. The utilization of PostgreSQL and MongoDB as databases allowed for the storage and retrieval of data in a structured and flexible manner.
- The project has achieved its objectives of providing a reliable and efficient system for users to search and retrieve car parts data from eBay. The system includes features such as category-based filtering, and data display in a user-friendly format. The system also includes error handling, data validation, and database integration for storing and retrieving data efficiently.
- Throughout the development of this project, several challenges were faced, including handling dynamic web pages, managing data integrity, and dealing with changes in the eBay website structure. However, through the use of appropriate technologies, such as Selenium, Beautiful Soup, Pandas, Psychopg2, and Numpy, these challenges were successfully addressed.
- Furthermore, the project has identified potential enhancements, such as implementing user authentication, real-time data updates, advanced search filters, recommendation system, and mobile application, among others, which could further enhance the system's functionality and usability.
- Overall, this project has provided valuable experience in web scraping, data handling, and database integration using Python and related libraries, and has the potential to be further extended and enhanced in the future. It has met the project objectives and has the potential to benefit users.

7. Bibliography

- Documentation of Selenium (Python) - Selenium documentation: <https://www.selenium.dev/documentation/en/>
- Documentation of Beautiful Soup - Beautiful Soup documentation: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Documentation of Pandas - Pandas documentation: <https://pandas.pydata.org/pandas-docs/stable/>
- Documentation of Psychopg2 - Psychopg2 documentation: <http://initd.org/psycopg/docs/>
- Documentation of Numpy - Numpy documentation: <https://numpy.org/doc/>
- Documentation of PostgreSQL - PostgreSQL documentation: <https://www.postgresql.org/docs/>
- Documentation of MongoDB - MongoDB documentation: <https://docs.mongodb.com/>
- Python official documentation - Python documentation: <https://docs.python.org/>
- GitHub - Open-source code hosting and collaboration platform: <https://github.com/>