



TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
PULCHOWK CAMPUS

## CV ANALYZER

SUBMITTED BY:

Bigyapti Bashyal (076BCT016)

Bivek Shrestha (076BCT020)

Ishani Malla (076BCT028)

SUBMITTED TO:

Department of Electronics and Computer Engineering

Lalitpur, Nepal

UNDER SUPERVISION OF

Prof. Lok Nath Regmi

UNDER MENTORSHIP OF

Prof. Bishal Dev Pandey

March, 2023

## Acknowledgement

We would like to extend our heartfelt thanks to our advisors for their invaluable guidance and unwavering support throughout this project. We are immensely grateful to the teachers, staff, and seniors who generously contributed their time and expertise to make this project a success.

We owe a debt of gratitude to our advisors, without whom this project would not have been possible. We would also like to express our appreciation to the Department of Electronics and Computer Engineering, Institute of Engineering, and Pulchowk Campus for their support.

We would like to acknowledge the efforts of our supervisor, Prof. Lok Nath Regmi, for his exceptional guidance and support throughout the project and our mentor, Bishad Pandey, for his invaluable input and feedback throughout the project.

We are also deeply grateful to our friends and family members who provided us with unwavering support and encouragement throughout this project, as well as to the seniors who generously shared their knowledge and expertise with us.

## Abstract

CV Analyzer is a tool designed to provide insights into job seekers' resumes, making it easier for them to apply for jobs. The tool analyzes the candidate's CV using Natural Language Processing (NLP) algorithms and provides a summary of their skills, experiences, and achievements. This also predicts the suitable job for the user. Additionally, it highlights areas that require improvement, such as gaps in employment or lack of relevant skills.

The platform also allows job administrators to post job listings and rank the best CVs based on the analysis provided by the tool. This helps employers to quickly identify the most suitable candidates for their job openings.

Overall, CV Analyzer is an efficient tool for both job seekers and employers, as it simplifies the job application process and enables recruiters to find the right candidate quickly and effectively.

## List of Figures

1. Pie Chart Showing Category Distribution in Dataset
2. Bar Chart showing Dataset Category Values and Category Count
3. Word Count Graph
4. Cleaned Data from Dataset
5. Model Architecture Diagram
6. Use Case Diagram
7. Admin Activity Diagram
8. User Activity Diagram
9. Login Page
10. User Page
11. Prediction Page
12. Job Vacancy Page (User)
13. Admin Job Vacancy Post Page
14. Job Vacancy Page (Admin)

## Table of Content

<b>Acknowledgement</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>1. Introduction</b>	<b>7</b>
1.1. Problem Statement	7
1.2. Objectives	8
<b>2. Literature Review</b>	<b>8</b>
2.1. NLTK (Natural Language Toolkit)	9
2.2. spaCy	9
<b>3. Theory</b>	<b>9</b>
3.1. Natural Language Processing (NLP)	9
3.2. Machine Learning (ML)	10
3.3. Data mining	10
<b>4. Software and Library Requirements</b>	<b>11</b>
4.1. Flask	11
4.2. MongoDB	12
4.3. Bootstrap	13
4.4. Custom CSS	13
4.5. NLTK	14
4.6. PDFMiner	14
4.7. Sklearn	14
4.8. KNN Algorithm	14
<b>5. Methodology</b>	<b>15</b>
5.1. Model training	15
5.1.1. Data collection	15
5.1.1.1 Dataset description:	16
5.1.2. Preprocessing	17
5.1.3. Model training	17
5.1.3.1. Model Architecture Diagram	18
5.1.3.2. KNN Algorithm	18
5.2. System Diagrams	19
5.2.1. Use Case Diagram	19
5.2.2. Activity Diagram	20
5.2.2.1. Admin Activity Diagram	20
5.2.2.2. User Activity Diagram	21
5.3. Application	22
5.3.1. Frontend	22
5.3.2. Backend	22

5.3.3. UI Interface	22
<b>6. Result and Analysis</b>	<b>23</b>
6.1. Output	23
<b>7. Epilogue</b>	<b>26</b>
7.1. Conclusion	26
7.2. Future Enhancement	26
<b>References</b>	<b>27</b>

# 1. Introduction

CV Analyzer is a cutting-edge tool designed to provide comprehensive insights into job seekers' resumes, allowing them to showcase their skills and experiences effectively. The platform employs advanced Natural Language Processing (NLP) algorithms that analyze the candidate's CV and extract relevant information.

CV Analyzer helps job seekers understand their CV's strengths and weaknesses, providing them with valuable insights into their job application process. The platform identifies gaps in employment, lack of relevant skills, and other areas that require improvement. This allows job seekers to make necessary adjustments to their CV and improve their chances of landing their dream job.

For job administrators, CV Analyzer provides a powerful tool for posting job listings and ranking the best CVs based on the analysis provided by the tool. This makes the hiring process more efficient and effective, allowing employers to find the right candidate quickly.

The platform also provides job seekers with personalized suggestions to enhance their CV, such as highlighting the skills and experiences that match the job requirements. This allows job seekers to tailor their resumes to each job they apply for, making them more competitive and increasing their chances of success.

CV Analyzer is a versatile tool that can be used by job seekers of all levels, from entry-level to senior executives. The platform is user-friendly and easy to navigate, allowing job seekers to create an attractive and professional-looking CV in a matter of minutes.

In conclusion, CV Analyzer is a game-changing platform that simplifies the job application process for job seekers and employers alike. With its advanced NLP algorithms, personalized suggestions, and efficient job ranking system, CV Analyzer helps job seekers land their dream job and employers find the right candidate quickly and effectively.

## 1.1. Problem Statement

CV Analyzers can help to solve several problems that may arise during the recruitment process, such as:

- The traditional resume screening process is time-consuming and error-prone.
- The manual review of numerous resumes can be overwhelming and tedious.

- Human bias is prevalent in the traditional screening process, leading to subjective decisions.
- Subjective decisions can lead to the exclusion of qualified candidates and the selection of less qualified ones.
- The quality of the hiring process is impacted by the traditional screening process.
- There is a need for a more efficient and objective method of screening resumes.
- The CV analyzer tool aims to automate the resume screening process, saving time, reducing errors, and eliminating human bias.
- The tool allows admin to post job opportunities and screen through all resumes.
- The recruitment process becomes more streamlined and effective with the CV analyzer tool.

## 1.2. Objectives

- Use CV to predict the most suitable job sectors for the candidate.
- Check if the candidate's CV matches the given job requirements.
- Rank candidates accordingly based on their CV scores and job requirements.
- Automate the resume screening process to reduce errors and save time.
- Eliminate human bias in the resume screening process.
- Provide recruiters with insights and analytics about the resume pool.
- Reviewing resume using Machine learning model

## 2. Literature Review

Research on CV analysis has been an active area of study in recent years, with many researchers exploring various methods for extracting information from resumes and using that information to predict job candidates' qualifications and suitability for a given job.

One common approach is to use natural language processing (NLP) techniques to extract information from resumes, such as education, work experience, skills, and qualifications. Researchers have proposed various NLP methods, including information extraction, named entity recognition, and sentiment analysis, to extract information from resumes. These methods



have been shown to be effective in extracting relevant information from resumes and in predicting job candidates' qualifications and suitability for a given job.

There are several popular libraries and frameworks available for natural language processing (NLP) in various programming languages. Some of the most popular include:

### 2.1. NLTK (Natural Language Toolkit)

NLTK is a widely-used library for NLP in Python. It includes modules for tokenization, stemming, tagging, parsing, semantic reasoning, and wrappers for industrial-strength NLP libraries.

### 2.2. spaCy

spaCy is a library for advanced natural language processing in Python. It's fast and efficient, and it's designed specifically for production use. It includes pre-trained models for a variety of languages and tasks, including named entity recognition, part-of-speech tagging, and dependency parsing.

## 3. Theory

### 3.1. Natural Language Processing (NLP)

A subfield of Artificial Intelligence that focuses on enabling machines to understand human language and communicate with people in natural language. Natural Language Processing (NLP) is a branch of artificial intelligence that deals with the interaction between computers and humans using natural language. It involves the development of algorithms and computer programs that can analyze, understand, and generate human language.

NLP is used in a wide range of applications, including language translation, sentiment analysis, chatbots, speech recognition, and text summarization. It involves several techniques such as syntactic and semantic analysis, machine learning, deep learning, and data mining.

NLP has become increasingly important in the age of big data, as there is a vast amount of text data available on the internet, social media, and other sources. By analyzing this data, NLP can help organizations gain insights into customer behavior, identify trends, and improve decision-making.

### 3.2. Machine Learning (ML)

Machine Learning (ML) is a subset of artificial intelligence that involves the use of statistical algorithms and mathematical models to enable computers to learn from data without being explicitly programmed. The goal of ML is to enable computers to automatically learn and improve from experience, without being explicitly programmed to do so.

There are three main types of ML algorithms: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the algorithm is trained on labeled data, meaning the input and output values are known, and the goal is to learn a function that can accurately predict the output for new inputs. In unsupervised learning, the algorithm is trained on unlabeled data, meaning only the input values are known, and the goal is to identify patterns or structure in the data. In reinforcement learning, the algorithm learns through trial and error by interacting with an environment and receiving feedback in the form of rewards or punishments.

ML is used in a wide range of applications, including image and speech recognition, natural language processing, recommendation systems, fraud detection, and predictive analytics. ML algorithms can help organizations automate tasks, improve decision-making, and gain insights from large datasets.

### 3.3. Data mining

Data mining techniques are used to extract useful information from large datasets by discovering patterns and relationships within the data. Here are some common data mining techniques used to extract useful information:

1. Association Rule Mining: This technique is used to find relationships between items in a dataset. For example, if customers who buy product A are likely to buy product B, this technique can help identify that relationship.

2. Clustering: This technique groups similar data points together based on their characteristics. For example, clustering can be used to group customers based on their purchasing behavior.
3. Classification: This technique is used to classify data into predefined categories. For example, classification can be used to predict whether a customer will purchase a product based on their past behavior.
4. Regression: This technique is used to predict a numerical value based on other variables in the dataset. For example, regression can be used to predict the price of a house based on its size, location, and other factors.
5. Neural Networks: This technique is used to simulate the functioning of the human brain and learn from data. It is particularly useful for image recognition and speech recognition.
6. Decision Trees: This technique is used to visualize the decision-making process and identify the most important factors that influence the outcome. For example, decision trees can be used to identify the factors that influence whether a customer will buy a product.

## **4. Software and Library Requirements**

### **4.1. Flask**

Flask is a lightweight, open-source web framework written in Python. It is designed to be simple and easy to use, making it a popular choice for developing web applications quickly.

Flask provides a wide range of features and extensions that can be used to build a variety of web applications, including RESTful APIs, web services, and full-stack web applications. Some of the key features of Flask include:

1. Routing: Flask provides a simple and intuitive way to define URL routes and map them to specific functions or views.
2. Templating: Flask includes a powerful templating engine that makes it easy to generate dynamic HTML content.

3. Debugging: Flask provides a built-in debugger that helps developers identify and fix errors quickly.
4. Flask-WTF: Flask-WTF is an extension that provides integration with the WTForms library, making it easy to create and validate web forms.
5. Flask-SQLAlchemy: Flask-SQLAlchemy is an extension that provides integration with the SQLAlchemy ORM, making it easy to work with databases in Flask applications.
6. Flask-RESTful: Flask-RESTful is an extension that provides support for building RESTful APIs in Flask.
7. Flask is also highly customizable and can be extended with a wide range of third-party libraries and tools. Overall, Flask is a versatile and powerful web framework that can be used to build a variety of web applications quickly and easily.

## 4.2. MongoDB

MongoDB is a popular open-source NoSQL database that stores data in a flexible, JSON-like format known as BSON. It is designed to be scalable, flexible, and fast, making it a popular choice for modern web and mobile applications.

Some of the key features of MongoDB include:

1. Flexible data model: MongoDB's document-based data model allows for easy storage of complex data structures and supports dynamic schema changes.
2. Horizontal scalability: MongoDB is designed to scale horizontally across multiple servers, making it easy to handle large datasets and high traffic loads.
3. Indexing and querying: MongoDB supports a wide range of indexing options and provides a powerful query language that allows for fast and efficient retrieval of data.
4. Automatic sharding: MongoDB supports automatic sharding, which allows data to be distributed across multiple servers for improved performance and scalability.
5. High availability: MongoDB provides built-in replication and failover support, ensuring that data is always available even in the event of hardware failures or other issues.

6. Rich ecosystem: MongoDB has a large and active community of developers and provides a wide range of tools and integrations, making it easy to work with other technologies and platforms.

#### 4.3. Bootstrap

Bootstrap is a popular open-source framework for building responsive, mobile-first web applications. It provides a set of CSS, JavaScript, and HTML templates and components that make it easy to create modern and professional-looking web interfaces. Some of the key features of Bootstrap include:

1. Responsive design: Bootstrap provides a grid system and responsive design classes that make it easy to create layouts that adapt to different screen sizes and devices.
2. Pre-built components: Bootstrap includes a wide range of pre-built UI components, such as buttons, forms, tables, and navigation menus, that can be easily customized and used in web applications.
3. Customizable styling: Bootstrap provides a set of variables and mixins that make it easy to customize the styling of the framework to match the branding and design of the application.
4. JavaScript plugins: Bootstrap includes a number of JavaScript plugins, such as modals, carousels, and tooltips, that enhance the functionality of web interfaces.
5. Large community: Bootstrap has a large and active community of developers who contribute to the framework and provide support through forums and other resources.

#### 4.4. Custom CSS

Custom CSS, or Cascading Style Sheets, is a powerful tool that allows web developers to customize the appearance of HTML elements on a web page. CSS works by defining rules that specify how HTML elements should be styled, such as their color, font, size, position, and more.

#### 4.5. NLTK

NLTK (Natural Language Toolkit) is a comprehensive open-source library for natural language processing (NLP) in Python. It provides a suite of tools and resources for working with human language data, including tokenization, part-of-speech tagging, named entity recognition, sentiment analysis, and more. With its extensive collection of pre-built corpora, models, and tools, NLTK is a popular choice for researchers, educators, and developers working with NLP applications. It also includes a range of machine learning algorithms for training and fine-tuning models on custom data, making it a versatile and flexible tool for a wide range of NLP tasks. Overall, NLTK is a powerful and user-friendly library that has played a significant role in advancing the field of NLP.

#### 4.6. PyPDF2

PyPDF2 is a Python library that provides tools for extracting information from PDF documents. It is designed to be efficient, flexible, and easy to use, and can be used to extract text, images, and other data from PDF files.

#### 4.7. Sklearn

Scikit-learn, also known as sklearn, is a popular open-source machine learning library for Python. It is designed to be simple, efficient, and accessible, making it easy for developers and data scientists to build and deploy machine learning models.

#### 4.8. KNN Algorithm

The K Nearest Neighbor algorithm is a nonparametric algorithm, which means it makes no data assumptions. It's also known as a lazy learner algorithm since it doesn't learn from the training set right away; instead, it saves the dataset and uses it to categorize it later. During the training phase, the KNN algorithm simply saves the dataset, and when it receives new data, it classifies it into a category that is quite similar to the new.

## 5. Methodology

### 5.1. Model training

- The prediction model uses a machine learning approach to predict a job candidate's suitability for a given job based on their resume.
- The model takes in a PDF file of the candidate's resume as input.
- The PDF file is first processed using the PyPDF2 library to extract the text from the file.
- The extracted text is then cleaned using regular expressions to remove special characters, line breaks, and extra whitespace.
- The cleaned text is transformed into a set of numerical features using a word vectorization technique, which converts each word in the text into a vector of numerical values.
- The feature set is then fed into the trained machine learning model, which predicts the most suitable job for the candidate based on their qualifications and experience.
- The model uses a classification algorithm to predict the job category, with each category corresponding to a specific type of job.
- The predicted job category is then returned to the user as output.
- The model has been trained on a large dataset of job descriptions and resumes to ensure accurate predictions.
- Creating a web application using flask incorporating the above model.

The model we propose has four steps:

#### 5.1.1. Data collection

The first step is to collect a dataset of resumes that will be used to train and test the CV Analyzer. The data set will be obtained from kaggle.

### 5.1.1.1 Dataset description:

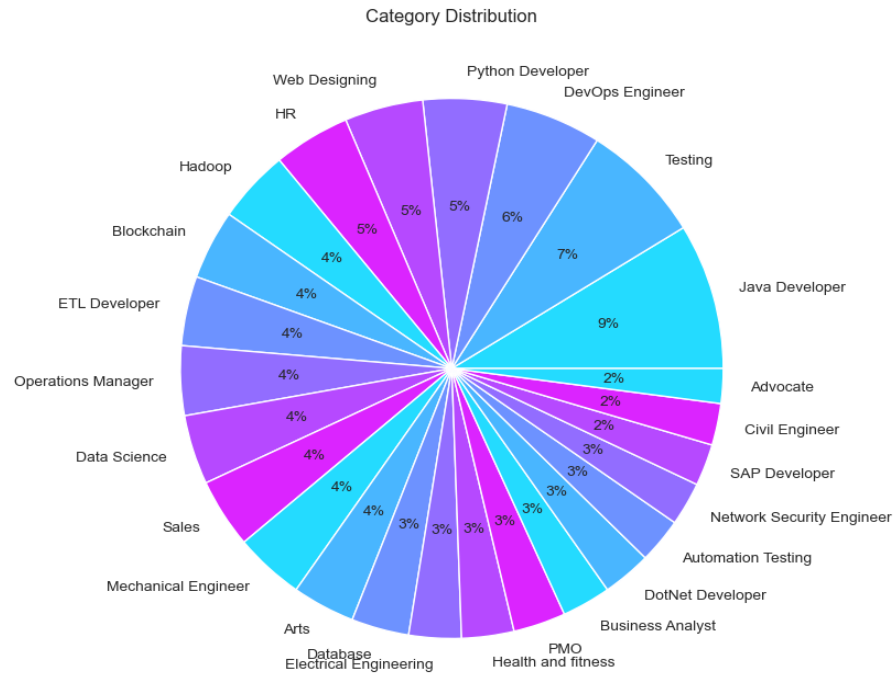


Figure 1: Pie Chart Showing Category Distribution in Dataset

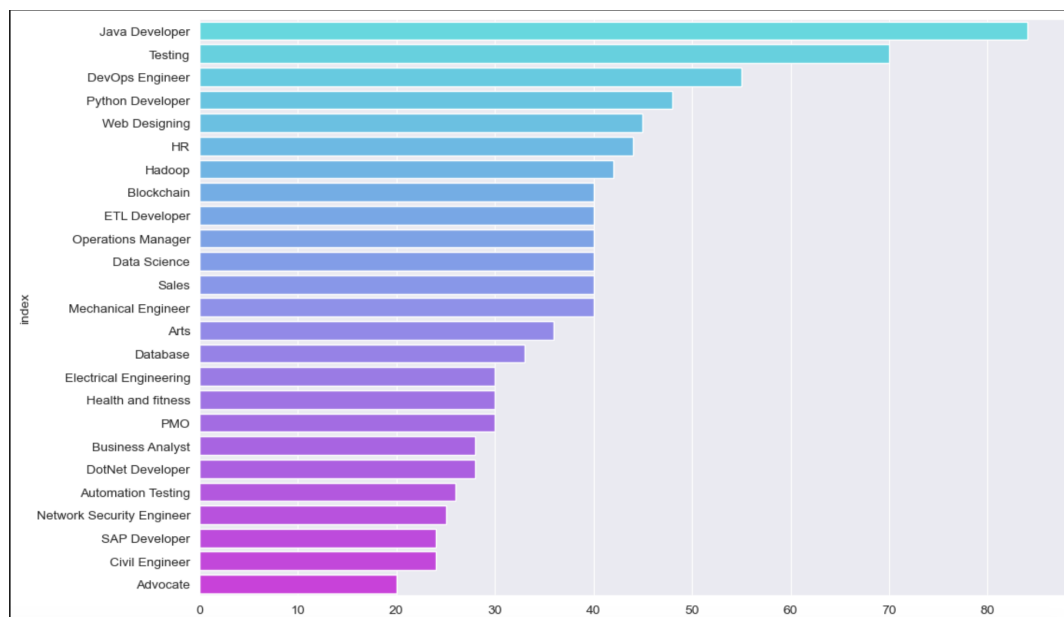


Figure 2: Bar Chart showing Dataset Category Values and Category Count



### 5.1.2. Preprocessing

The next step is to preprocess the resumes by cleaning, normalizing, and formatting the data. To extract information from the candidates resume we use PyResparser and PDFminer



Figure 3: Word Count Graph

Category	Resume	clean text
0	6	Skills * Programming Languages: Python (pandas...
1	6	Education Details 'inMay 2013 to May 2017 B E...
2	6	Areas of Interest Deep Learning, Control Syste...
3	6	Skills & R & Python & SAP HANA & Table...
4	6	Education Details 'in MCA YMCAUST, Faridab...
...	...	...
957	23	Computer Skills: & Proficient in MS office (...)
958	23	& Willingness to accept the challenges. & ...
959	23	PERSONAL SKILLS & Quick learner, & Eagerne...
960	23	COMPUTER SKILLS & SOFTWARE KNOWLEDGE MS-Power ...
961	23	Skill Set OS Windows XP/7/8/8.1/10 Database MY...

Figure 4: Cleaned Data from Dataset

### 5.1.3. Model training

The model's concept will be trained with existing data gathered from the Kaggle open platform. The first model, either K-Nearest Neighbor and will help us predict what kind of job role our resume is best suited for, while the second model, cosine similarity, will check the similarity of user's input to posted job vacancies.

At the frontend, the candidate uploads their resume; the resume is then passed to the resume parser, which is a pipeline of NLP algorithms that extracts important information from the resume; and

finally, adding more value to the overall extracted data from vectors and providing it to the Machine learning Model for tagging.

For job requirement match check the following steps are implemented:

- Pdf data collection
- Job requirement data collection
- TF-IDF vectorization of data
- Feeding the TF-IDF vectors into the cosine similarity algorithm

TF-IDF vectorization

IDF is calculated as follows where  $t$  is the term (word) we are looking to measure the commonness of and  $N$  is the number of documents ( $d$ ) in the corpus ( $D$ ). The denominator is simply the number of documents in which the term,  $t$ , appears in.

$$idf(t, D) = \log \left( \frac{N}{count(d \in D : t \in d)} \right)$$

TF-IDF is the importance of a term is inversely related to its frequency across documents. TF gives us information on how often a term appears in a document and IDF gives us information about the relative rarity of a term in the collection of documents. By multiplying these values together we can get our final TF-IDF value.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

The higher the TF-IDF score the more important or relevant the term is; as a term gets less relevant, its TF-IDF score will approach 0.

## Cosine similarity algorithm

Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. We can measure the [similarity between two sentences in Python](#) using Cosine Similarity. In cosine similarity, data objects in a dataset are treated as a vector. The formula to find the cosine similarity between two vectors is –

$$\text{Cos}(x, y) = x \cdot y / \|x\| * \|y\|$$

Cosine similarity in textual data is used to compare the similarity between two text documents or tokenized texts. So in order to use cosine similarity in text data, the raw text data has to be tokenized at the initial stage, and from the tokenized text data a similarity matrix has to be generated which can be passed on to the cosine similarity metrics for evaluating the similarity between the text document.

### 5.1.3.1. Model Architecture Diagram

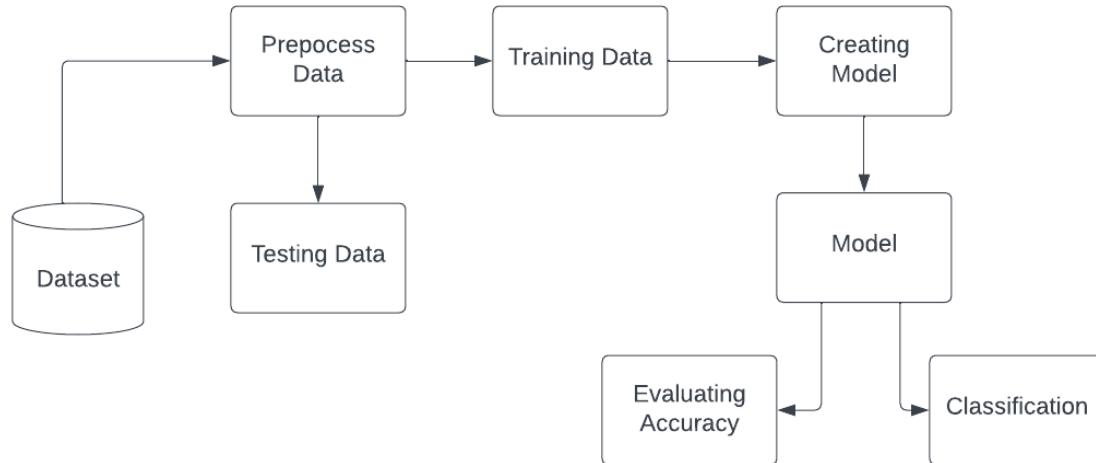


Figure 5: Model Architecture Diagram

### 5.1.3.2. KNN Algorithm

The K Nearest Neighbor algorithm is a non- parametric algorithm, which means it makes no data assumptions. It's also known as a lazy learner algorithm since it doesn't learn from the training set right away; instead, it saves the dataset and uses it to categorize it later. During the training phase, the KNN algorithm simply saves the dataset, and when it receives new data, it classifies it into a category that is quite similar to the new data

---

#### **Algorithm:** The Traditional KNN Algorithm

---

**Input:** the training set  $D$ , test object  $x$ , category label set  $C$

**Output:** the category  $c_x$  of test object  $x$ ,  $c_x$  belongs to the  $C$

```

1  begin
2    for each  $y$  belongs to  $D$  do
3      calculate the distance  $D(y, x)$  between  $y$  and  $x$ 
4    end for
5    select the subset  $N$  from the data set  $D$ ,
      the  $N$  contains  $k$  training samples which are the  $k$ 
      nearest neighbors of the test sample  $x$ 
6    calculate the category of  $x$ :
      
$$c_x = \arg \max_{c \in C} \sum_{y \in N} I(c = \text{class}(y))$$

7  end
  
```

## 5.2. System Diagrams

### 5.2.1. Use Case Diagram

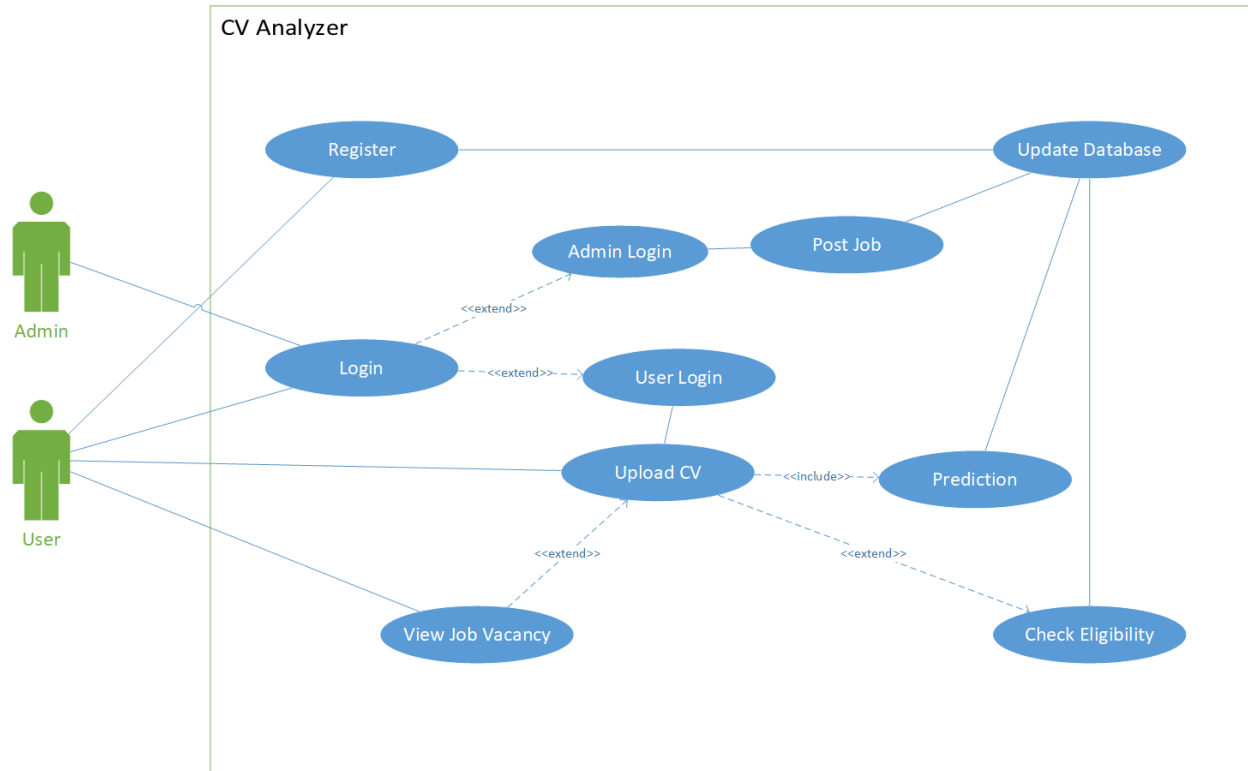


Figure 6: Use Case Diagram

## 5.2.2. Activity Diagram

### 5.2.2.1. Admin Activity Diagram

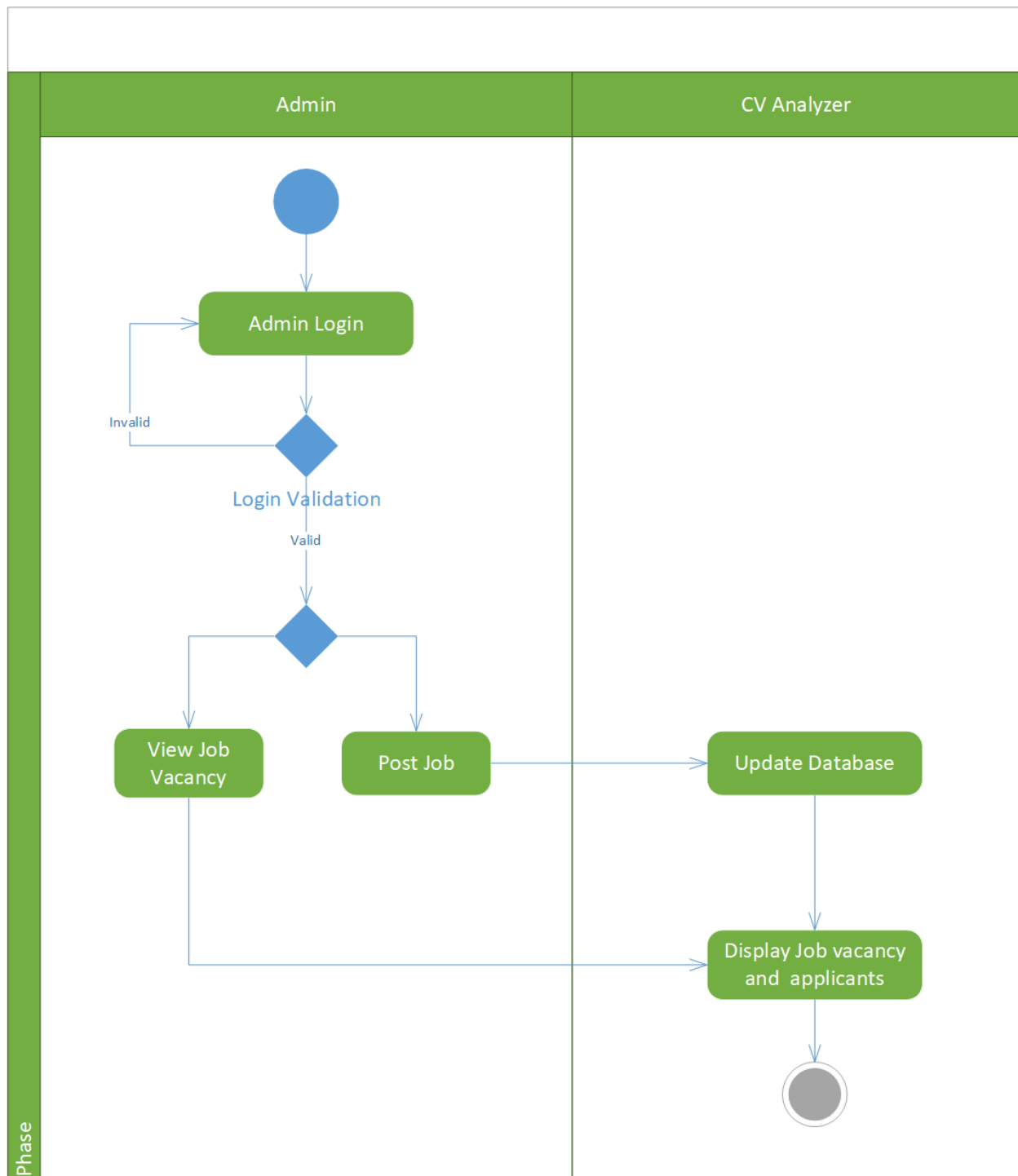


Figure 7: Admin Activity Diagram

### 5.2.2.2. User Activity Diagram

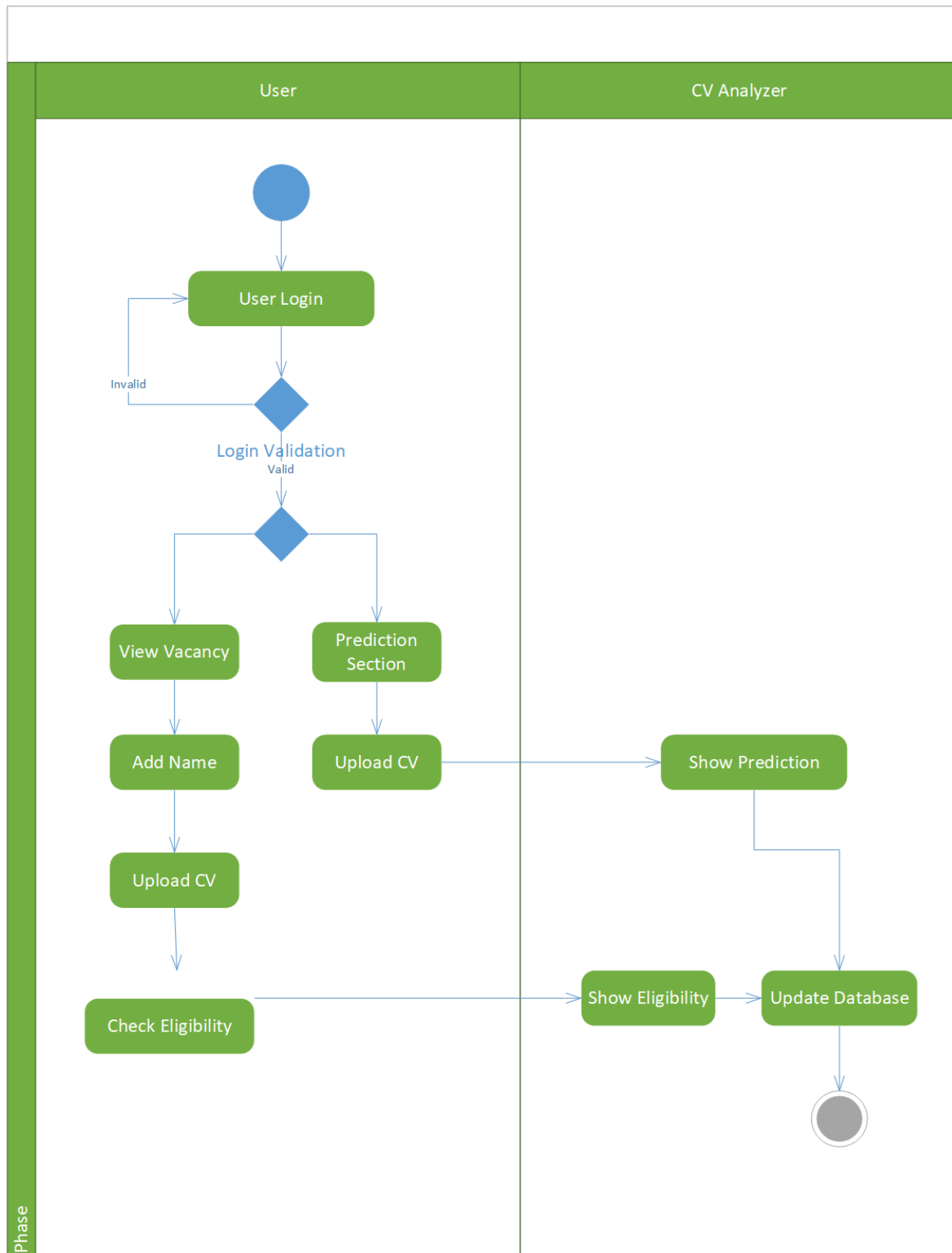


Figure 8:User Activity Diagram

## 5.3. Application

### 5.3.1. Frontend

The UI was built using HTML, CSS and Bootstrap.

### 5.3.2. Backend

The backend server for the CV Analyzer application was implemented using the Flask web framework in Python. Whenever a new resume is submitted for analysis, the server processes the resume and extracts relevant information using NLP and ML algorithms. The server then provides personalized suggestions to the job seeker for improving their resume, based on the analysis results.

Additionally, when a new job opening is posted, the server uses job classification and candidate matching algorithms to rank the submitted resumes based on their relevance to the job requirements. The server then sends the ranked list of resumes to the job administrator for review.

### 5.3.3. UI Interface


The web user interface (UI) for the CV Analyzer application was designed with Bootstrap and custom CSS to provide a responsive and user-friendly experience for job seekers and administrators.



## 6. Result and Analysis

### 6.1. Output

CV Analyzer User Login Admin Login



Username  
hi

Password  
[password field with eye icon]

Login

Don't have an account? [Register](#)

Figure 9: Login Page

CV Analyzer Predict Job Category View Job Vacancies Logout

### Unlock the Power of Your CV with our Intelligent Analyzer

Upload Your CV

Choose File no file selected

**ANALYZE CV**




Figure 10: User Page

CV Analyzer

Predict Job Category View Job Vacancies Logou

Prediction:

Data Science

Resume:

**Data Scientist**  
**ROBERT SMITH**

Phone: (123) 456 78 99  
Email: info@qwikresume.com  
Website: www.qwikresume.com  
LinkedIn: linkedin.com/qwikresume  
Address: 1737 Marshville Road, Alabama.

**Objective**  
Data Scientist with PhD in Physics and 1+ industrial experience. Two years of working experience in Data Analysis team of LIGO Scientific Collaboration [\$3M Special Breakthrough Prize winner of 2016]. Over ten years of successful research experience in both theoretical and computational physics. Strong problem-solving and analytical skills. Advanced programming proficiency. Certified in Data Analysis and Machine Learning.

**Skills**  
Data Mining, Data Analysis, Machine Learning, Python, R, MATLAB, Sphinx, LaTeX, Mathematica, Maple, GIT, CVS, HTCondor.

**Work Experience**  
**Data Scientist**  
**ABC Corporation** - May 1994 - May 2005  
Accerted in determining client needs, deliverable, deadline, estimation and feasibility for

Figure 11: Prediction Page

CV Analyzer

Predict Job Category View Job Vacancies Logou

Job Vacancies

Sr. No	Title	Description	Location	Check Eligibility
1	python developer	good experience with pandas numpy data science tensorflow keras model training machine learning natural language processing	kathmandu nepal	Name: <input type="text"/> Upload CV: <input type="button" value="Choose File"/> no file selected <input type="button" value="Check Eligibility"/>
2	data science	python pandas tensorflow data pytorch	nepal	Name: <input type="text"/> Upload CV: <input type="button" value="Choose File"/> no file selected <input type="button" value="Check Eligibility"/>
3	data science	As a data scientist, you should have a strong foundation in mathematics, statistics, and computer science. You should be proficient in programming languages such as Python and R, and have experience with SQL and data	nepal	Name: <input type="text"/>

Figure 12: Job Vacancy Page (User)

CV Analyzer

View Job Vacancies Logout

### Post Job Vacancy

Job Title

Job Description

Location

Post Job Vacancy

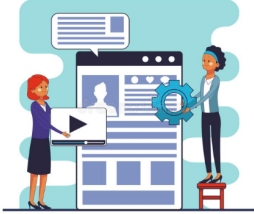


Figure 13: Admin Job Vacancy Post Page

CV Analyzer

Post Job Vacancies Logout

Job Vacancies

Job Title	Job Description	Location	Candidates Applied Ranking
python developer	good experience with pandas numpy data science tensorflow keras model training machine learning natural language processing	kathmandu nepal	1. DATA (Similarity: 42.0%) 2. DATA (Similarity: 42.0%) 3. name (Similarity: 28.999999999999996%) 4. name (Similarity: 28.999999999999996%) 5. sth (Similarity: 28.999999999999996%) 6. adage (Similarity: 22.0%) 7. adage (Similarity: 22.0%) 8. DATA (Similarity: 21.0%) 9. IOS (Similarity: 21.0%) 10. ram (Similarity: 11.76083881682571%) 11. Sam (Similarity: 11.76083881682571%) 12. hi (Similarity: 8.0%) 13. asd (Similarity: 8.0%) 14. Bigyapli (Similarity: 7.969708913291729%) 15. hell (Similarity: 7.969708913291729%) 16. Justin (Similarity: 5.661385170722979%) 17. asd (Similarity: 4.0%) 18. from (Similarity: 4.0%) 19. Hari (Similarity: 3.6735917918532257%)

Figure 14: Job Vacancy Page (Admin)

## 7. Epilogue

### 7.1. Conclusion

In conclusion, the CV Analyzer project offers an efficient and effective solution for analyzing resumes and matching candidates to job openings. The project utilizes advanced Natural Language Processing (NLP) and Machine Learning (ML) algorithms to extract relevant information from resumes and rank them based on their relevance to job requirements. The backend server is built using Flask, providing a lightweight and flexible architecture that is easily customizable and scalable. The web user interface is designed with Bootstrap and custom CSS, providing a responsive and user-friendly experience for job seekers and administrators.

The CV Analyzer project has the potential to significantly streamline the recruitment process, saving time and resources for both job seekers and employers. By providing personalized suggestions for improving resumes and matching candidates to job openings, the project offers a valuable tool for both job seekers and administrators.

### 7.2. Future Enhancement

Future work for the CV Analyzer project could include expanding the algorithmic capabilities for analyzing resumes and further customization of the user interface. Additionally, integration with other recruitment platforms and social media could enhance the project's reach and effectiveness. Here are some potential future enhancements for the CV Analyzer project:

1. Integration with other job boards: Integration with other job boards such as LinkedIn, Indeed, and Glassdoor could provide a wider pool of job openings for candidates and more diverse candidate pools for employers.
2. Additional algorithms for resume analysis: Developing additional algorithms for analyzing resumes could enhance the accuracy and effectiveness of the analysis. For example, sentiment analysis could provide insights into the candidate's attitude and personality traits.
3. Automated interview scheduling: Integrating with scheduling tools such as Calendly or Doodle could allow for automated scheduling of candidate interviews, saving time for both candidates and employers.

4. Personalized job recommendations: Developing algorithms for personalized job recommendations could enhance the user experience for candidates and increase their chances of finding the right job.
5. Analytics dashboard: Developing an analytics dashboard could provide job administrators with insights into the success of their job postings and candidate matching efforts.
6. Mobile app development: Developing a mobile app could provide a more convenient and accessible experience for job seekers who prefer to use their mobile devices for job searches.

These future enhancements could further improve the functionality and effectiveness of the CV Analyzer project and make it an even more valuable tool for both job seekers and employers.

## References

- [1] A. Kao and S. R. Poteet, "Text Mining and Natural Language Processing," 1st ed., CRC Press, 2007.
- [2] S. Raschka and V. Mirjalili, "Python Machine Learning," 3rd ed., Packt Publishing, 2019.
- [3] R. Arumugam, R. Shanmugamani, and B. Subramanian, "Hands-On Natural Language Processing with Python," Packt Publishing, 2018.
- [4] M. Grinberg, "Flask Web Development," 2nd ed., O'Reilly Media, Inc., 2018.
- [5] J. Lett, "Bootstrap 4 Quick Start," Packt Publishing, 2018.