Industry Project
Report
On

Word to CSV Converter & Career Predictor

Developed By: -
Manya Tripathi (20162121009)

Guided By:-
Prof. Umesh Lakhtariya (Internal)
Mr. Suraj Bhatnagar (External)

Submitted to
Faculty of Engineering and Technology
Institute of Computer Technology
Ganpat University

Year - 2024

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
Institute of Computer Technology

CERTIFICATE

This is to certify that the Industry Training and Internship by Manya Tripathi (Enrolment No.20162121009) of Ganpat University, towards the partial fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them at EHE Data Systems. The results/findings contained in this Internship have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of Internal Guide

Name & Signature of Head

Place: ICT – GUNI

Date:

ACKNOWLEDGEMENT

Industry Internship project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank Dr. Rohit Patel, Principal, ICT, and Prof. Dharmesh Darji, Head, ICT who gave us an opportunity to undertake this project. My grateful thanks to Prof. Umesh (Internal Guide) for his guidance. I am appreciative of the support and mentorship provided bythe EHE Data system's team, which has been instrumental in shaping my understanding of the industry. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Manya Tripathi (Enrollment No:20162121009)

INDEX

Week 1: Basics of File connections and File handling

1. **File Connection:**
    - Learn about mongodb , mysql and file system connection using python .

2. **File Handling:**
    - Complete python file handling module and python modules from w3schools and youtube.

Week 2: Learning About the Project, Playwright, and Web Automation

1. **Web Automation with Playwright**:
    - Understand the basics of Playwright for web automation.
    - Learn how to launch browsers, navigate pages, and interact with web elements.
    - Explore headless mode for efficient automation.
2. **Environment Setup:**
    - Install Python and set up a virtual environment.
    - Install necessary packages: **Playwright**, **Beautiful Soup**, and **Celery**.
    - Initialize a Flask project for the web interface.

Week 3: Web Scraping and HTML Parsing

1. Beautiful Soup for HTML Parsing:

- Dive deeper into Beautiful Soup.
- Learn how to extract relevant data from HTML pages.
- Practice scraping data from sample websites.
- Integration of Beautiful Soup for HTML parsing into the project.
- Exploration of advanced Playwright techniques to handle dynamic elements, frames, andcomplex scenarios.

Week 4: Selenium and Playwright Integration

1. Selenium vs. Playwright:

- Comparing Selenium and Playwright for web automation.
- Deciding to use playwright for the project.

2. Integrate Playwright:

- Exploration of the Playwright library for web automation.
- Acquisition of essential skills in Playwright, focusing on web page navigation and data extraction.

Week 5: Background Task Processing with Celery

1. **Introduction to Celery**:
    - Understanding asynchronous task processing.
    - Setting up Celery for background tasks.

2. Integrate Celery with Flask:

- Learning about Celery tasks for converting links to CSV files , and how to handle task queues andworkers.

Week 6: Creating the Web Interface with Flask

1. **Flask Web Framework**:
    - Learn Flask basics: routing, templates, and views.
2. **HTML, CSS, and JS:**
    - Designing a user-friendly web interface using HTML and CSS.
    - Implement file upload functionality.

Week 7: Working on word to CSV converter using react

1. Integrate mammoth Library:

- Installing mammoth library using npm in my React project.
- Updating the React code to use mammoth for processing Word documents.
- Test the integration by uploading a Word document.

2. Integrate file-saver Library:

- Install the file-saver library using npm.
- Update the React code to use file-saver for downloading CSV files.
- Test the integration by downloading a CSV file.

Week 8: File Handling

1. File Handling in Flask:

- Studied file handling mechanisms in Flask using request.files for file uploads.
- Understood how to validate and process file uploads from client requests.

2. Document Processing:

- Learned about the docx library for working with Microsoft Word documents in Python.
- Explored methods to read text data from Word documents using the Document class.

Week 9 : Flask Setup

1. **Initialized a Flask application and enabled CORS for handling requests from different origins.**
2. **File Processing Route:**
    - Implemented a route / to handle POST requests for processing Word files.
    - Validated file uploads and checked for the correct file format (.docx).
3. **Document Processing Functions:**
    - Developed functions to read text data from Word files and convert it to CSV format.
    - Utilized docx library for reading Word documents and csv module for CSV conversion.

Week 10 : Working on flask

1. Flask Route Handling:

- Explored advanced route handling techniques in Flask, such as route parameters and HTTP methods.
- Learned about request handling methods like GET, POST, and how to access request data.

2. Enhancing Error Handling:

- Studied strategies for improving error handling in Flask applications, including exception handling and response formatting.
- Implemented error handling mechanisms to provide informative error messages to clients.

3. File Download and Response Handling:

- Explored methods for serving files for download in Flask applications.
- Learned about send_file function for sending files as responses and setting appropriate MIME types.

Week 11 : Converting to CSV

1. CSV Preview Response:

- Enhanced the / route to provide a preview of the converted CSV data as JSON response.
- Implemented error handling for invalid file uploads and incorrect file formats.

2. CSV Download Route:

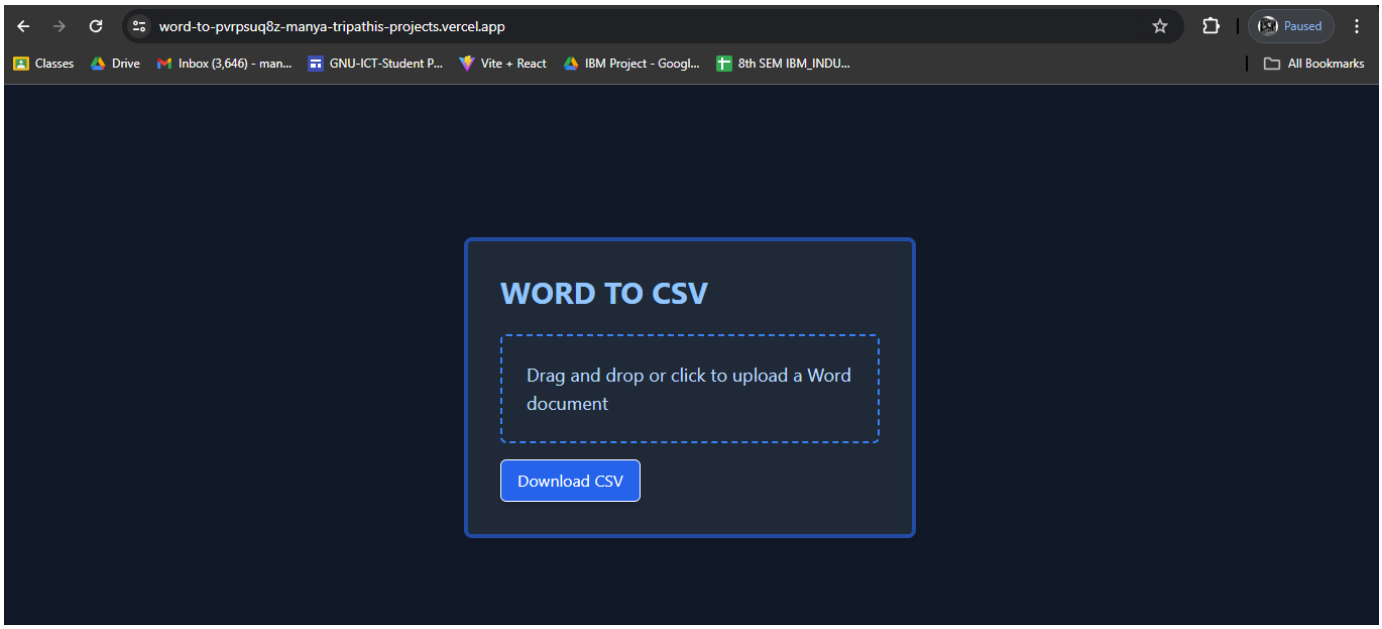
- Created a new route /download to handle POST requests for downloading the converted CSV file.
- Implemented response handling to send the CSV file as an attachment to client requests.

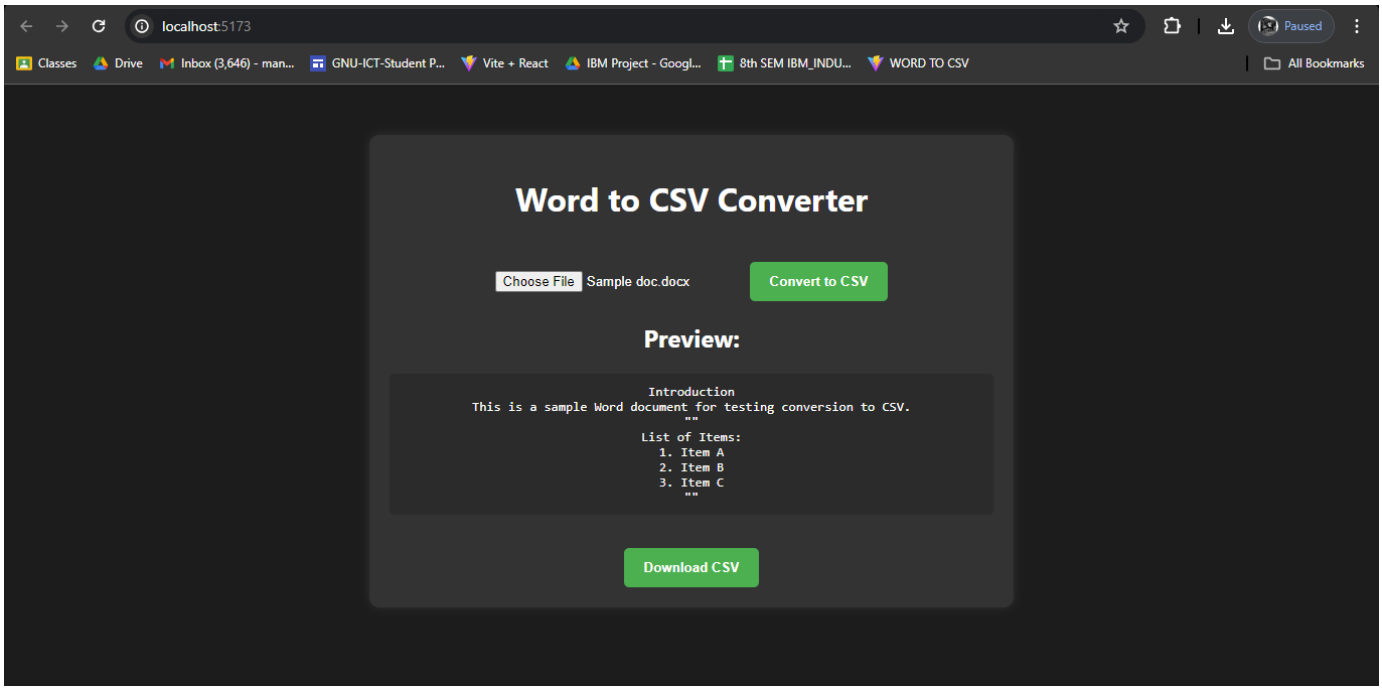
3. Testing and Debugging:

- Conducted testing of API endpoints using tools like Postman to ensure correct functionality.
- Debugged issues related to file processing, CSV conversion, and response handling.

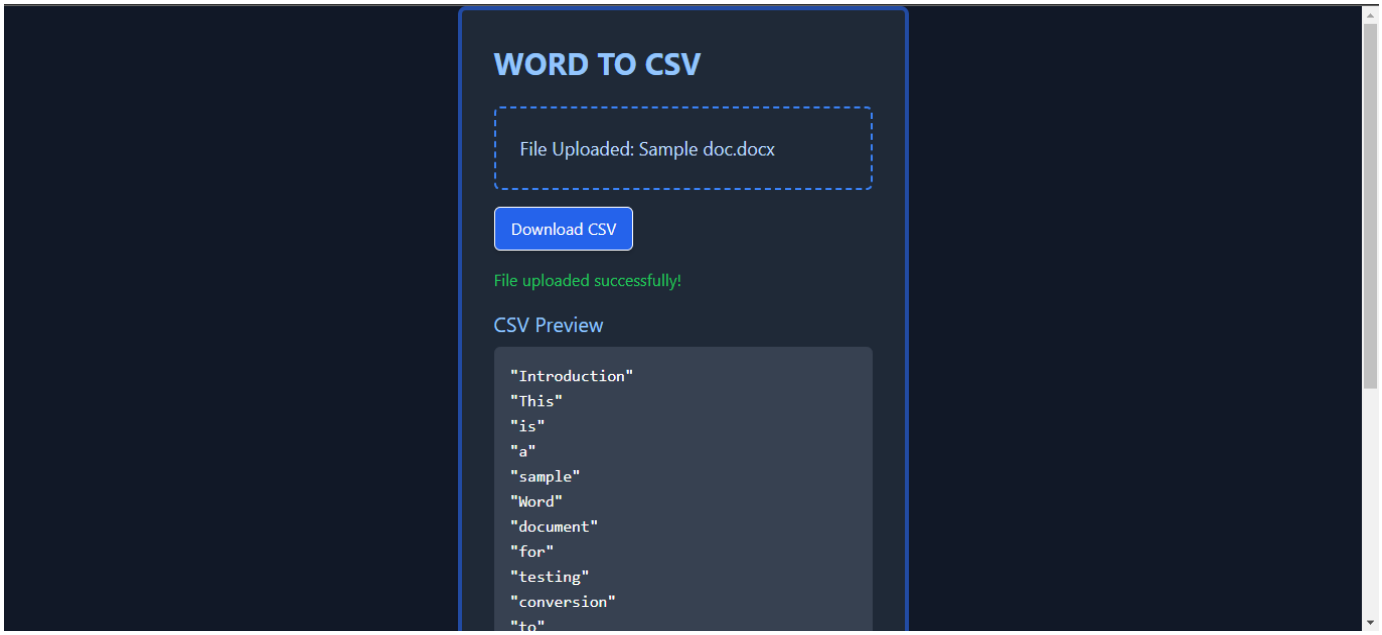
Screenshots of Word to CSV Converter –

Converter – 1





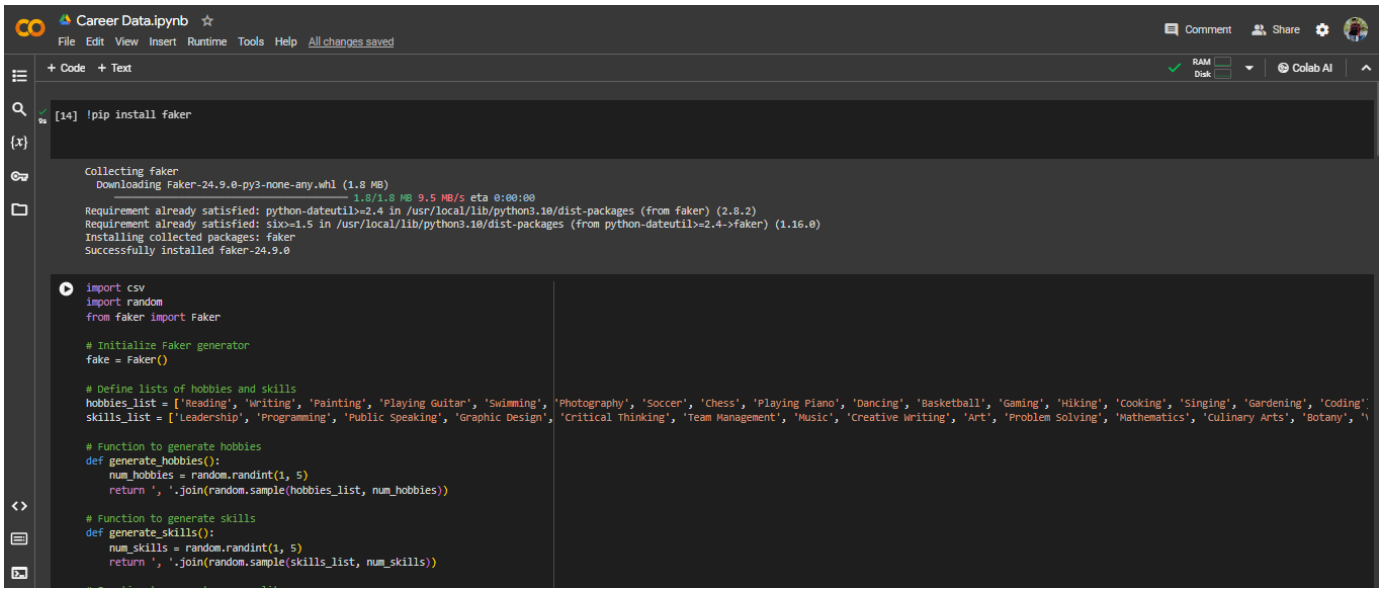
Converter – 2 : Using Python + Flask



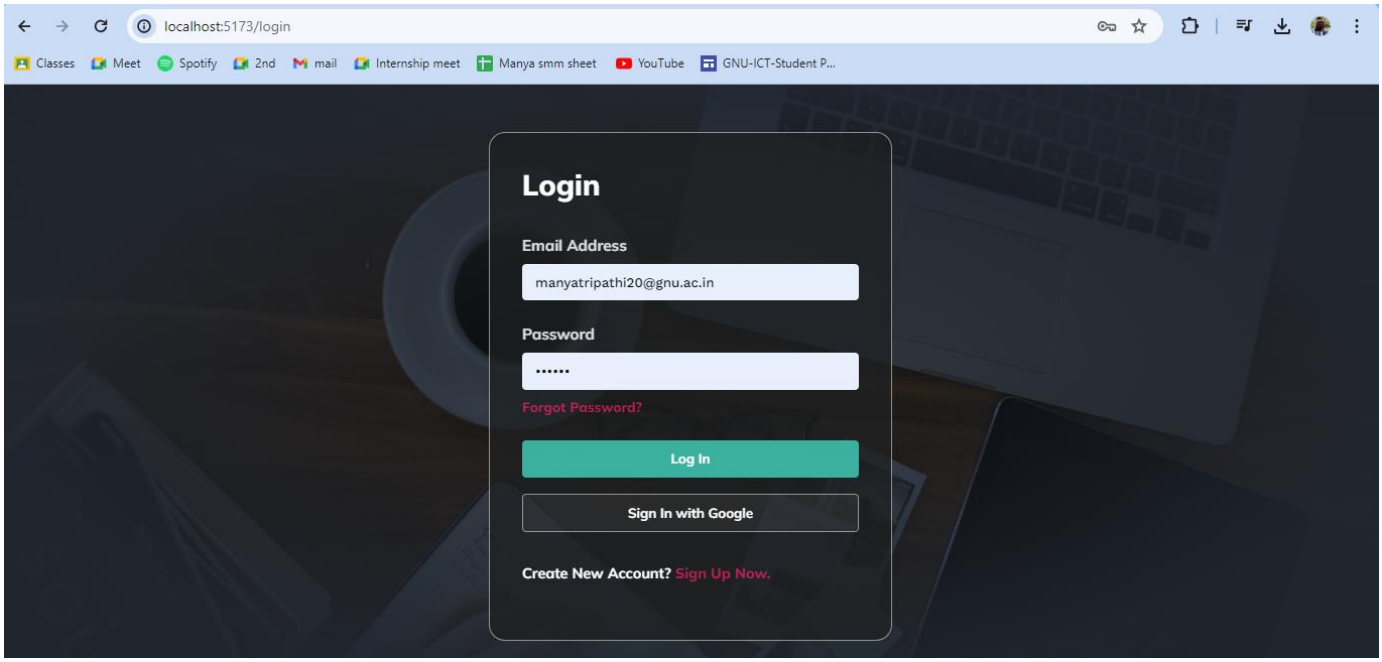
Week 15 - 16 : Career Prediction Project

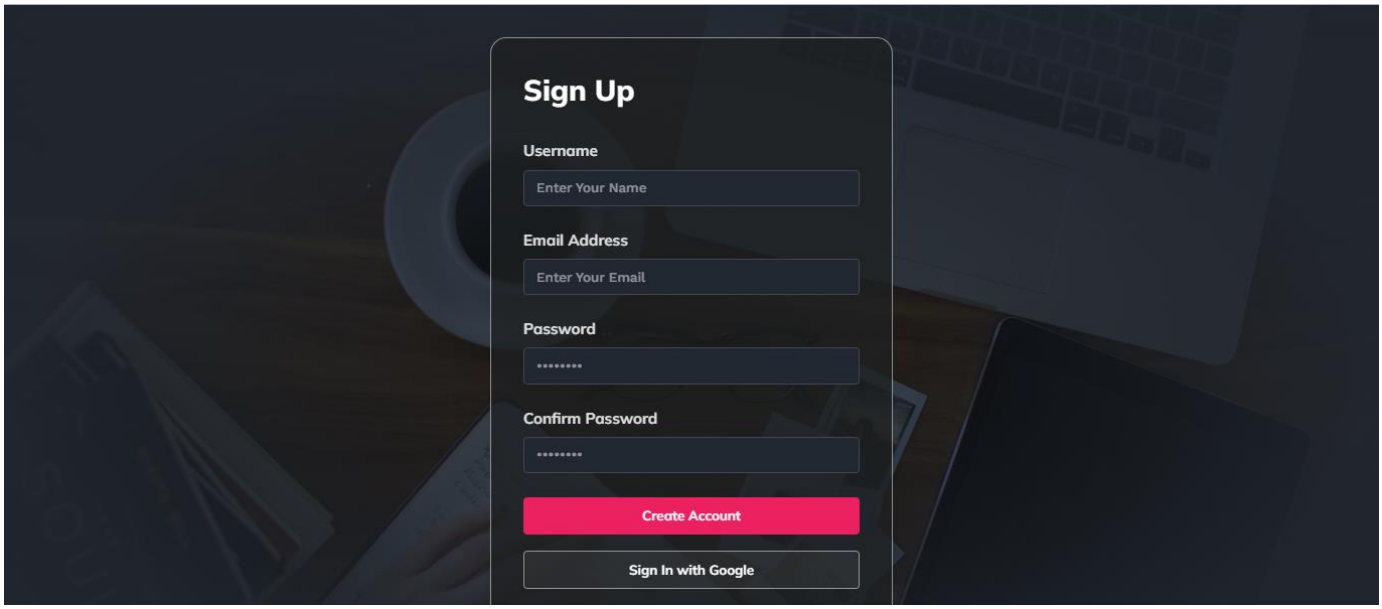
1. Finding suitable dataset for training model :

- Creating synthetic data using faker module in python
- Creating model and checking with different classifiers for predicting the career path.

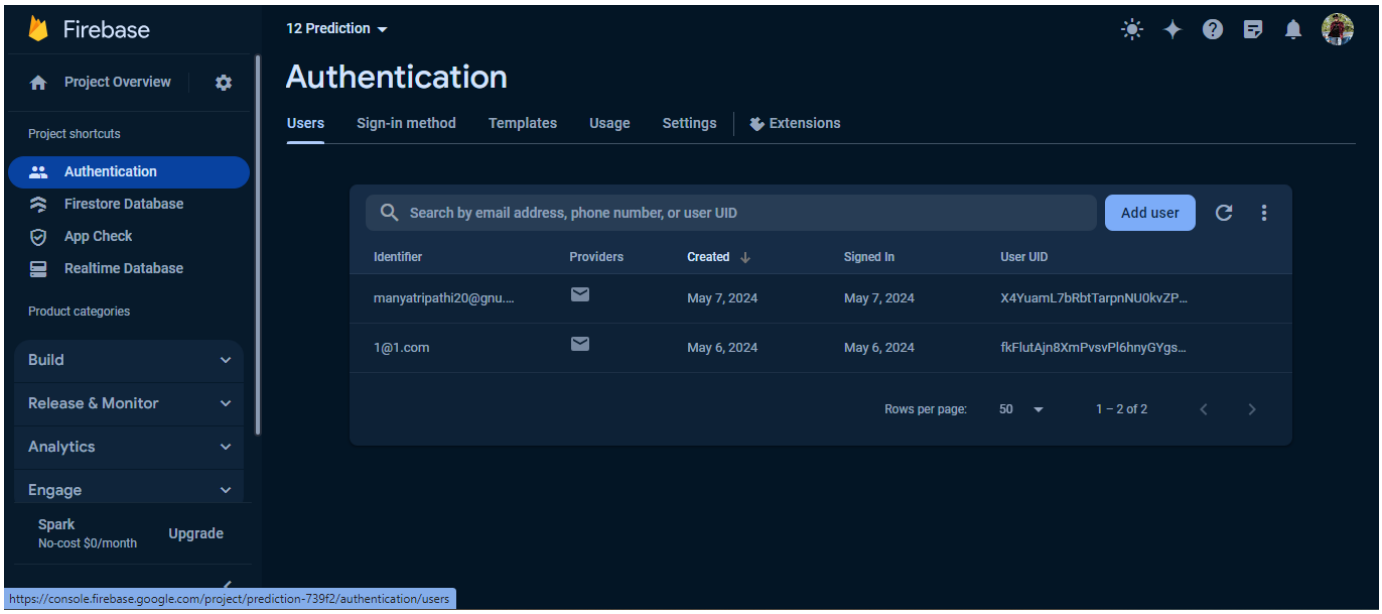


2. Creating the Login and Signup Page using firebase firestore for authentication and backend

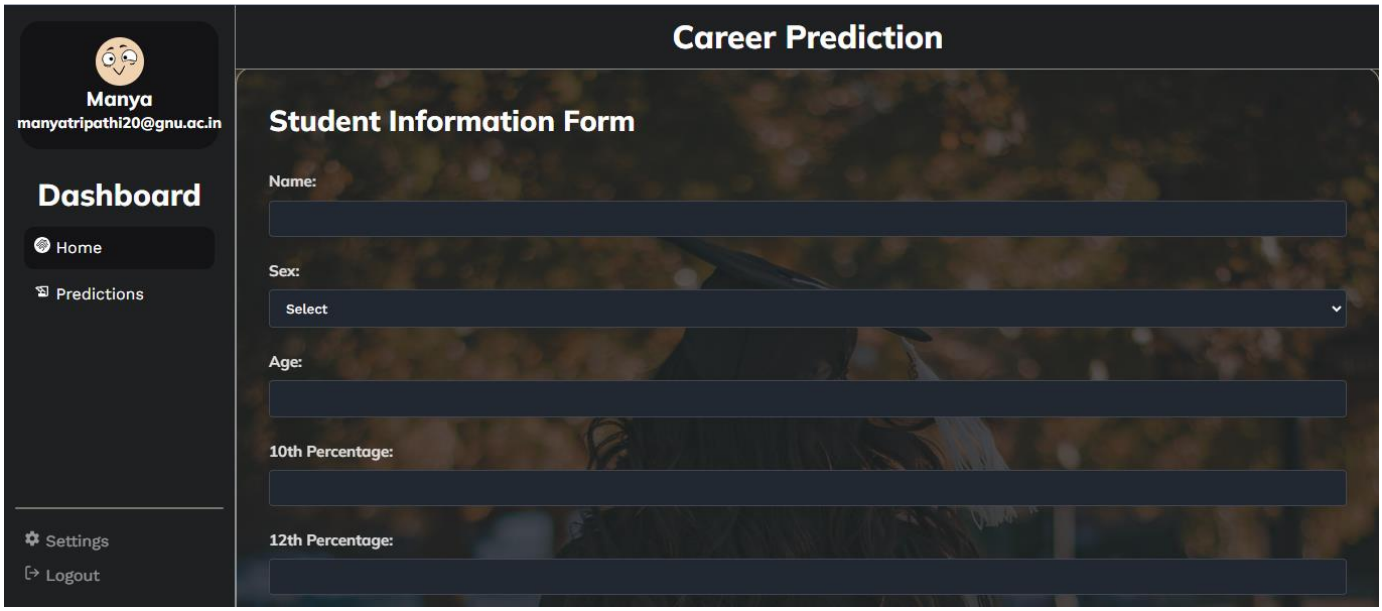




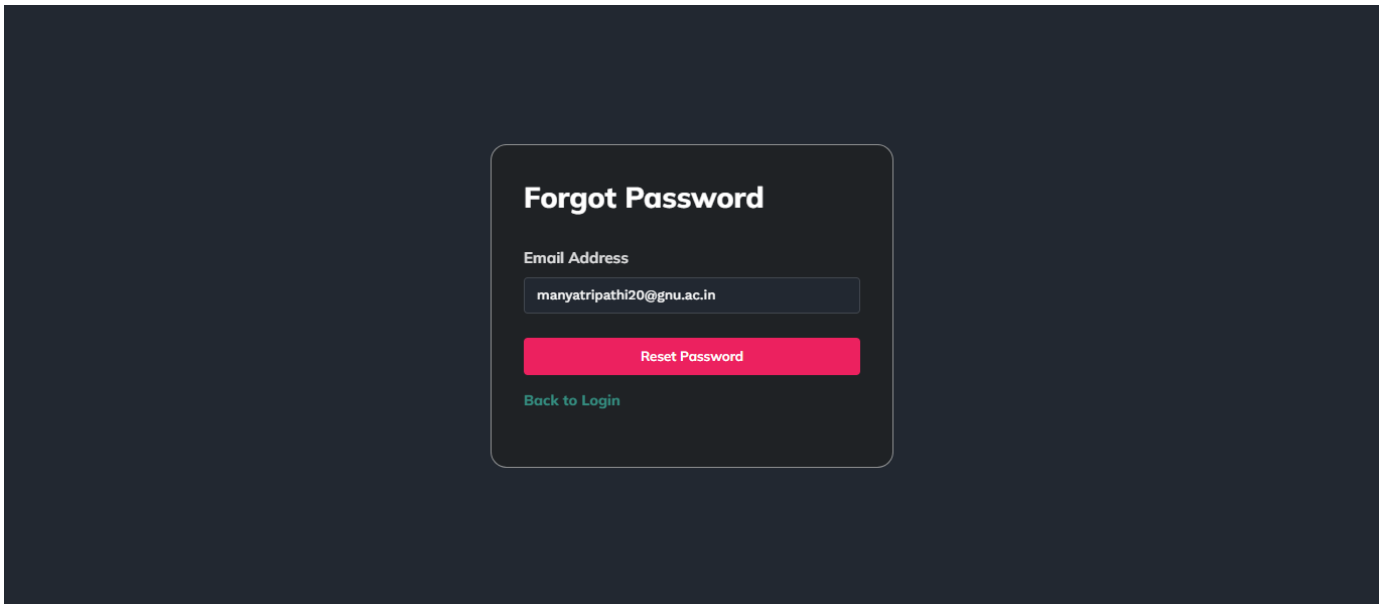
3. Connecting the login and signup to backend

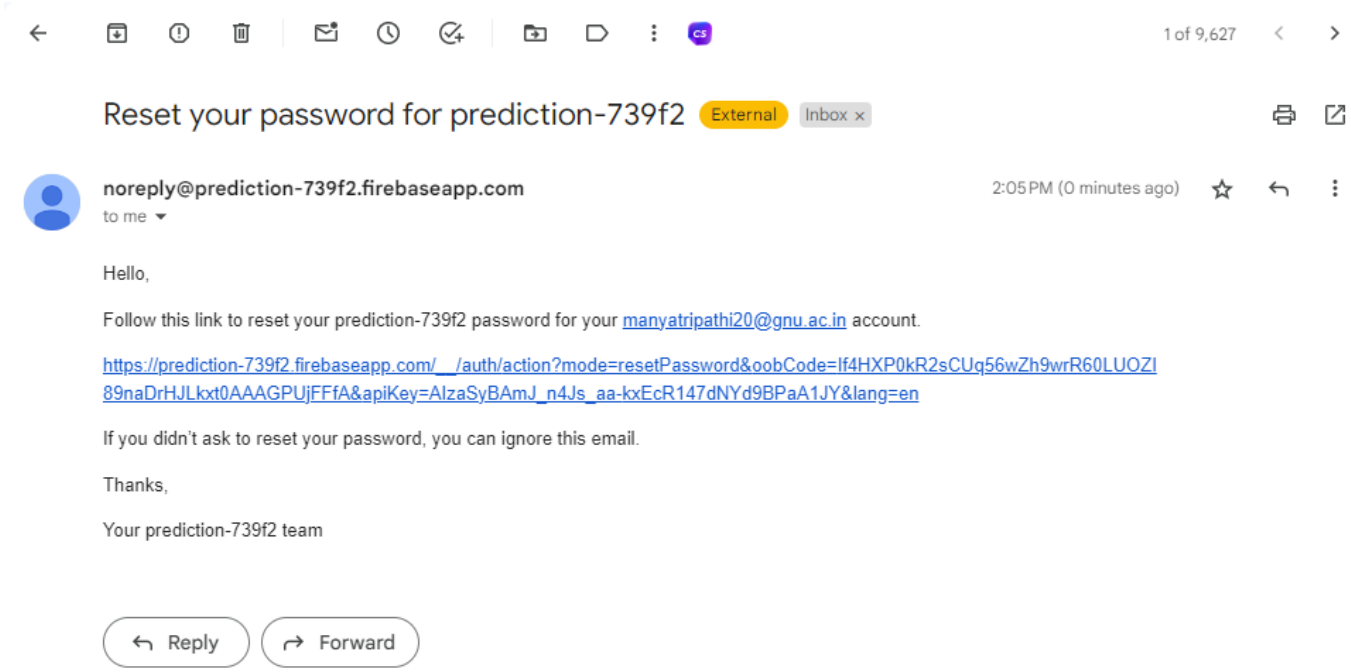


4. Creating the home page and the career prediction form



5. Adding the forgot password functionality.





Reset your password for prediction-739f2 External Inbox ×

1 of 9,627

noreply@prediction-739f2.firebaseapp.com
to me

2:05 PM (0 minutes ago)

Hello,

Follow this link to reset your prediction-739f2 password for your manyatripathi20@gnu.ac.in account.

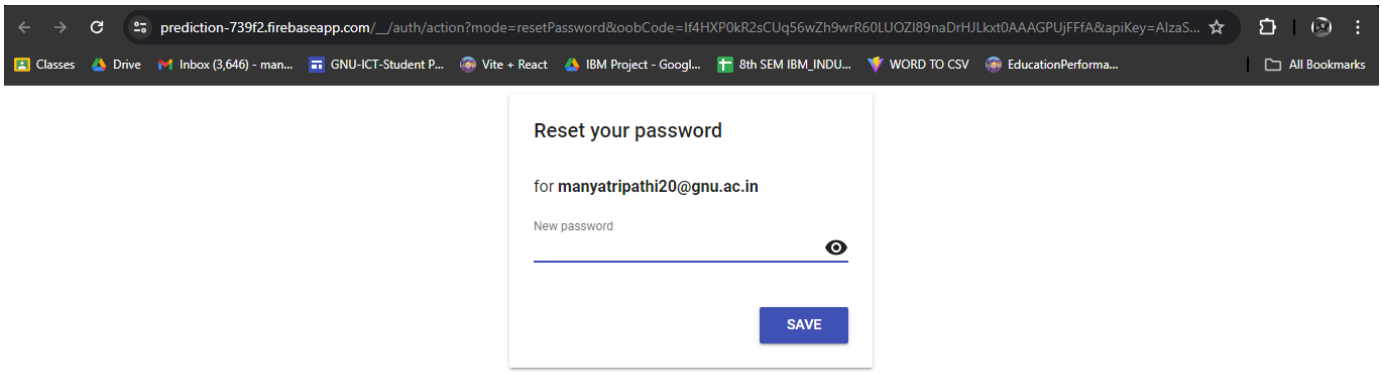
https://prediction-739f2.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=lf4HXP0kR2sCUq56wZh9wrR60LUOZI89naDrHJLkxt0AAAGPUjFFfA&apiKey=AIzaSyBAmJ_n4Js_aa-kxEcR147dNYd9BPaA1JY&lang=en

If you didn't ask to reset your password, you can ignore this email.

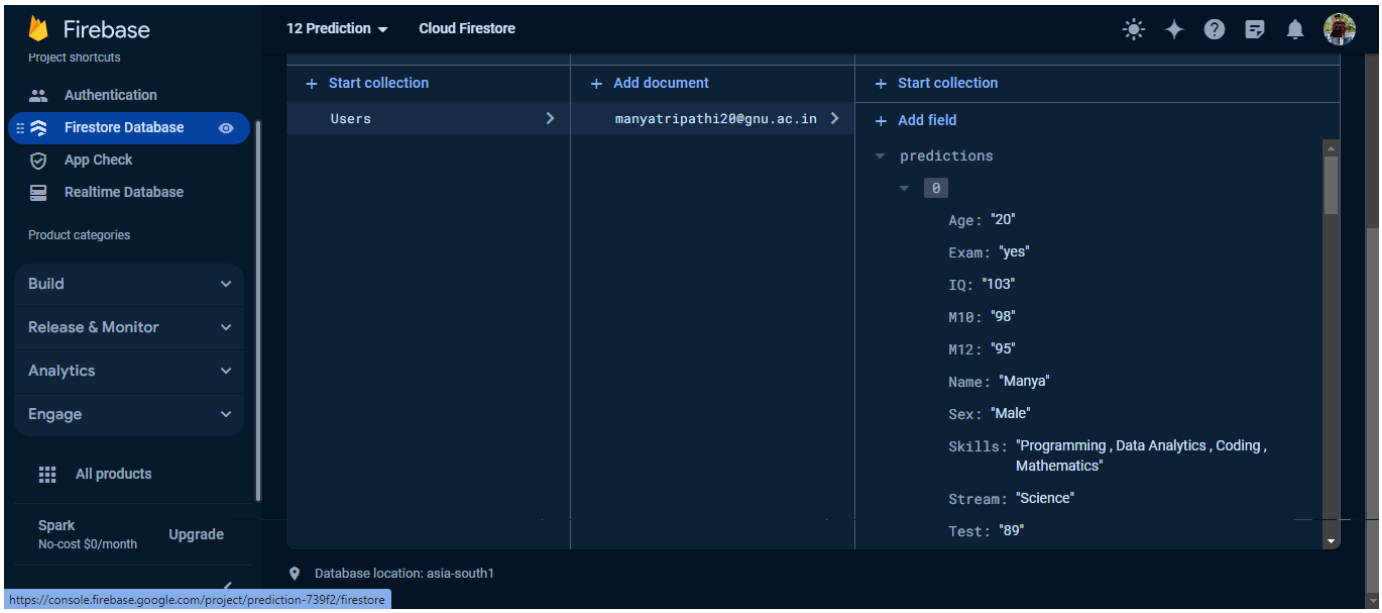
Thanks,

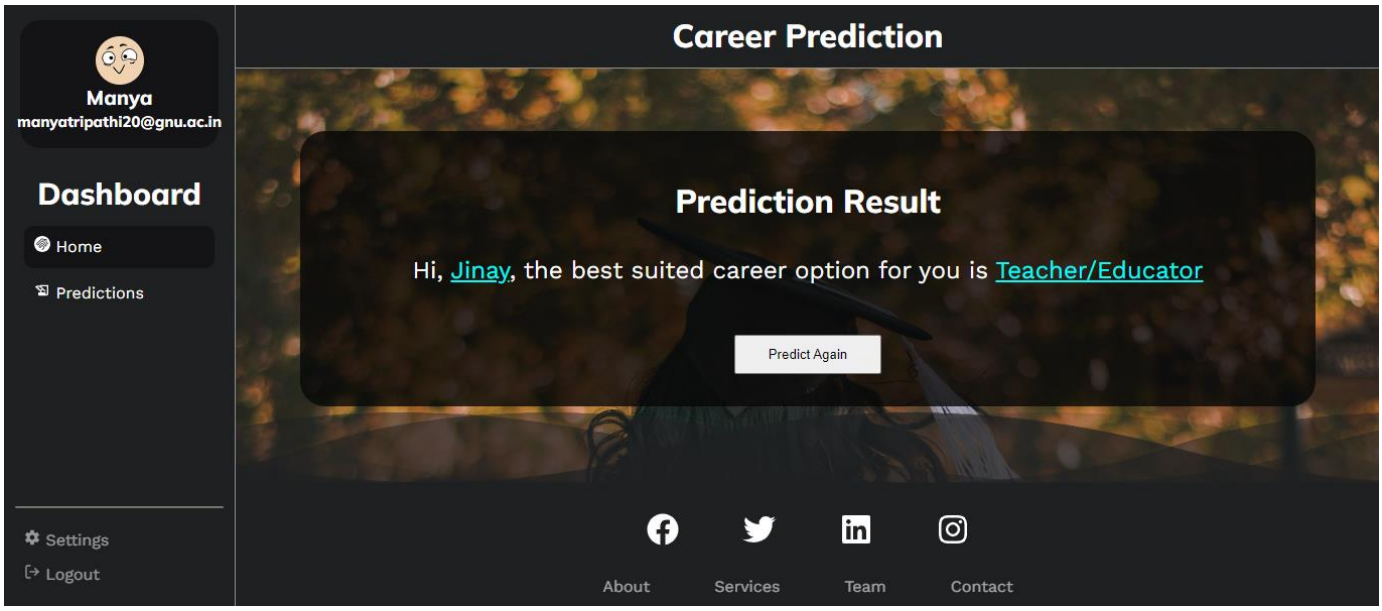
Your prediction-739f2 team

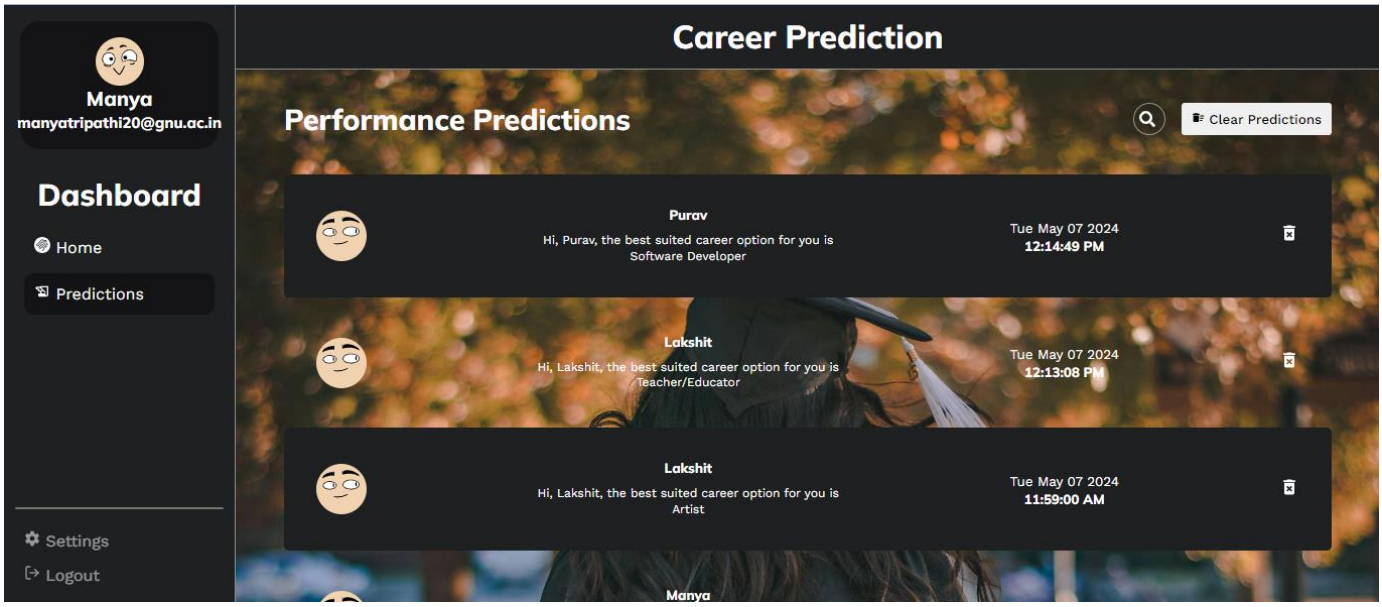
Reply Forward



6. Connecting the model with backend using flask, predicting the career option and storing the predictions.






Career Prediction
Manya
manyatripathi20@gnu.ac.in
Dashboard
Home
Predictions
Settings
Logout
Performance Predictions
Clear Predictions
Purav
Hi, Purav, the best suited career option for you is Software Developer
Tue May 07 2024
12:14:49 PM
Lakshit
Hi, Lakshit, the best suited career option for you is Teacher/Educator
Tue May 07 2024
12:13:08 PM
Lakshit
Hi, Lakshit, the best suited career option for you is Artist
Tue May 07 2024
11:59:00 AM
Manya

CONCLUSION

In 16 weeks of intensive learning, I acquired proficiency in file connections, handling, and web automation using Playwright. The internship commenced with mastering MongoDB, MySQL, and file system connections. Subsequent weeks focused on Playwright for web automation, including environment setup and integration into a Flask project. Advanced techniques in web scraping with Beautiful Soup and dynamic element handling were explored. The decision to integrate Playwright over Selenium marked a crucial phase. Background task processing using Celery and Flask integration followed. Currently, I am honing Flask basics and designing a user-friendly web interface using HTML, CSS, and JS, culminating in a well-rounded skill set.

In conclusion, the "Career Predictor " project has successfully developed a machine learning solution to assist students in making informed career decisions. By leveraging data on academic performance, skills, and interests, the project provides personalized career recommendations, aiding students in selecting suitable career paths aligned with their abilities and aspirations. The implemented models, including Random Forest, SVM, KNN, Decision Tree, Naive Bayes, and ANN, demonstrate promising accuracy in predicting career options.

REFERENCES

[1] Ramaswami, M., and R. Bhaskaran. "A CHAID based performance prediction model in educational data mining." arXiv preprint arXiv:1002.1144 (2010).

[2] Imran, Muhammad, Shahzad Latif, Danish Mehmood, and Muhammad Saqlain Shah. "Student academic performance prediction using supervised learning techniques." International Journal ofEmerging Technologies in Learning 14, no. 14 (2019).

[3] Trakunphutthirak, Ruangsak, and Vincent CS Lee. "Application of educational data mining approach for student academic performance prediction using progressive temporal data." Journalof Educational Computing Research 60, no. 3 (2022): 742-776.

Plagiarism Detector .net

May 07, 2024

Plagiarism Scan Report

Characters:7401	Words:1000
Sentences:58	Speak Time: 8 Min

Excluded URL	None

Content Checked for Plagiarism

Week 1: Basics of File connections and File handling 1. File Connection: o Learn about mongodb , mysql and file system connection using python . 2. File Handling: o Complete python file handling module and python modules from w3schools and youtube. 2 Week 2: Learning About the Project, Playwright, and Web Automation 1. Web Automation with Playwright: o Understand the basics of Playwright for web automation. o Learn how to launch browsers, navigate pages, and interact with web elements. o Explore headless mode for efficient automation. 2. Environment Setup: o Install Python and set up a virtual environment. o Install necessary packages: Playwright, Beautiful Soup, and Celery. o Initialize a Flask project for the web interface. 3 Week 3: Web Scraping and HTML Parsing 1. Beautiful Soup for HTML Parsing: o Dive deeper into Beautiful Soup. o Learn how to extract relevant data from HTML pages. o Practice scraping data from sample websites. o Integration of Beautiful Soup for HTML parsing into the project. o Exploration of advanced Playwright techniques to handle dynamic elements, frames, and complex scenarios. 4 Week 4: Selenium and Playwright Integration 1. Selenium vs. Playwright: o Comparing Selenium and Playwright for web automation. o Deciding to use playwright for the project. 2. Integrate Playwright: o Exploration of the Playwright library for web automation. o Acquisition of essential skills in Playwright, focusing on web page navigation and data extraction. Week 6: Creating the Web Interface with Flask o Learn Flask basics: routing, templates, and views. 2. HTML, CSS, and JS: o Designing a user-friendly web interface using HTML and CSS. o Implement file upload functionality. 7 Week 7: Working on word to CSV converter using react 1. Integrate mammoth Library: o Installing mammoth library using npm in my React project. o Updating the React code to use mammoth for processing Word documents. o Test the integration by uploading a Word document. 2. Integrate file-saver Library: o Install the file-saver library using npm. o Update the React code to use file-saver for downloading CSV files. o Test the integration by downloading a CSV file. 8 Week 8: File Handling 1. File Handling in Flask: o Studied file handling mechanisms in Flask using request.files for file uploads. o Understood how to validate and process file uploads from client requests. 2. Document Processing: o Learned about the docx library for working with Microsoft Word documents in Python. o Explored methods to read text data from Word documents using the Document class. 9 Week 9 : Flask Setup 1. Initialized a Flask application and enabled CORS for handling

requests from different origins. 2. File Processing Route: o Implemented a route / to handle POST requests for processing Word files. o Validated file uploads and checked for the correct file format (.docx). 3. Document Processing Functions: o Developed functions to read text data from Word files and convert it to CSV format. o Utilized docx library for reading Word documents and csv module for CSV conversion. 10 Week 10 : Working on flask 1. Flask Route Handling: o Explored advanced route handling techniques in Flask, such as route parameters and HTTP methods. o Learned about request handling methods like GET, POST, and how to access request data. 2. Enhancing Error Handling: o Studied strategies for improving error handling in Flask applications, including exception handling and response formatting. o Implemented error handling mechanisms to provide informative error messages to clients. 3. File Download and Response Handling: o Explored methods for serving files for download in Flask applications. o Learned about send_file function for sending files as responses and setting appropriate MIME types. 11 Week 11 : Converting to CSV 1. CSV Preview Response: o Enhanced the / route to provide a preview of the converted CSV data as JSON response. o Implemented error handling for invalid file uploads and incorrect file formats. 2. CSV Download Route: o Created a new route /download to handle POST requests for downloading the converted CSV file. o Implemented response handling to send the CSV file as an attachment to client requests. 3. Testing and Debugging: o Conducted testing of API endpoints using tools like Postman to ensure correct functionality. o Debugged issues related to file processing, CSV conversion, and response handling. Screenshots of Word to CSV Converter – Converter – 1 12 Converter – 2 : Using Python + Flask 13 Week 15 - 16 : Career Prediction Project 1. Finding suitable dataset for training model : o Creating synthetic data using faker module in python o Creating model and checking with different classifiers for predicting the career path. 2. Creating the Login and Signup Page using firebase firestore for authentication and backend 14 3. Connecting the login and signup to backend 15 4. Creating the home page and the career prediction form 5. Adding the forgot password functionality. 16 17 6. Connecting the model with backend using flask, predicting the career option and storing the predictions. 18 Page 19 CONCLUSION In 16 weeks of intensive learning, I acquired proficiency in file connections, handling, and web automation using Playwright. The internship commenced with mastering MongoDB, MySQL, and file system connections. Subsequent weeks focused on Playwright for web automation, including environment setup and integration into a Flask project. Advanced techniques in web scraping with Beautiful Soup and dynamic element handling were explored. The decision to integrate Playwright over Selenium marked a crucial phase. Background task processing using Celery and Flask integration followed. Currently, I am honing Flask basics and designing a user- friendly web interface using HTML, CSS, and JS, culminating in a well-rounded skill set. In conclusion, the "Career Predictor " project has successfully developed a machine learning solution to assist students in making informed career decisions. By leveraging data on academic performance, skills, and interests, the project provides personalized

career recommendations, aiding students in selecting suitable career paths aligned with their abilities and aspirations. The implemented models, including Random Forest, SVM, KNN, Decision Tree, Naive Bayes, and ANN, demonstrate promising accuracy in predicting career options.

Sources