# Design Overview for ***Resume Updater***

Name: Sachin Kharel

Student ID: 689206

# Summary of Program

The **Resume Updater** is a Python-based automation tool designed to update my personal website's resume section (resume.html) by extracting content from a .docx resume file. This program enables efficient, error-free, and consistent updates to online resumes directly from a centralised, editable document.

It reads structured content (e.g., Education, Experience, Projects) from the .docx file, parses it using predefined rules, converts it to valid HTML, replaces corresponding sections in the resume.html file, and finally pushes the changes to my GitHub repository using Git commands. This allows me to maintain my resume in Word format while ensuring my website’s resume page remains updated with minimal effort.

The project is of reasonable complexity, includes multiple Python files (modular structure), uses 3+ classes, Enums, functional decomposition, abstraction, file I/O, error handling, and subprocess automation for Git.

# Required Data Types

Table : **Resume Section (Data Class)** details

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| title | str | The headings of different section of the resume (e.g “Education”, “Experience”, “Skills”, etc) |
| content\_html | str | HTML representation of each section’s content |
| section\_type | SectionType | An enum representing the type of different sections |

Table 2: **Resume Update Request (Data Class)** details

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| docx\_path | str | File path of the updated .docx resume |
| html\_path | str | Path to resume.html file for website |
| sections | List[ResumeSection] | Sections extracted and converted into HTML |

Table 3: **Git Change (Data Class)** details

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| file\_path | str | Path to HTML file being updated |
| commit\_msg | Str | The commit message used for Git commit |
| pushed | bool | whether the change has been pushed to Github |

Table 1: **File Type (Enum)** details

|  |  |  |
| --- | --- | --- |
| Member name | Member value | Notes |
| DOCX | “docx” | Microsoft Word Document |
| HTML | “html” | HTML document for website display |

Table 2: **Section Type (Enum)** details

|  |  |  |
| --- | --- | --- |
| Member name | Member value | Notes |
| EDUCATION | “education” | Education related resume section |
| EXPERIENCE | “experience” | Work experience related section |
| SKILLS | “skills” | Skills related resume section |
| PROJECTS | “projects” | Projects related resume section |
| LEADERSHIP | “leadership” | Leadership related resume section |

# Overview of Program Structure

|  |  |  |  |
| --- | --- | --- | --- |
| Module or class | Name | Return type | Description |
| main.py | main() | None | Main code of the program, handles user input, runs the update with error handling |
| file\_utils.py | validate\_file\_path() | bool | Checks if provided path exists and file extension is valid |
| parser.py | parse\_docx() | List[ResumeSection] | Parses .docx into structured ResumeSection objects |
| html\_updater.py | update\_html() | bool | Updates resume.html file by replacing target sections with new content |
| git\_utils.py | commit\_and\_push() | bool | Commits and push changes to GitHub |
| enums.py | File Type, Section Type | Enum | Enums used to define the types and section types |
| section.py | Resume Section, Resume Update Request, Git Change | dataclass | Data representations used throughout the project |