**Team 11 Skills and Resources Audit**

**Project overview**

Project name : Document Matcher

Project description : Develop a web application which automates the process of detecting common RSID codes in Microsoft Word documents to highlight evidence of collusion, or cheating.

**Objective of this document:**

* Assess skills needed to complete the project
* What to learn, and where is the resource materials from

**Skills required**

Programming skills

File handling

* Handling .docx documents, changing them to .zip to access and extract xml data

XML parsing

* Parsing and manipulating XML data, particularly RSID codes

Python: Used for data processing, flask backend

* Data processing and creating flask backend
* Error handling, input validation

Implementing security measures

* Implementing HTTPS, CRSF protection, and XSS prevention

Data analysis

RSID comparison

* Comparing RSID save codes to find commonality between saves and documents

Testing

User acceptance testing

* Project tested by client
* Test cases based on real world scenarios

System testing

* Test complete / integrated application to ensure met requirements

Functional testing

* Testing functions to ensure they work as intended

Integration testing

* Test interaction between different parts of application

Unit testing

* Writing test cases for components of code to test correctness

Performance testing

* Test applications performance under different scenarios

Web development (Full stack)

Frontend

* Frontend skills to develop web interface
* HTML, Jinja, CSS, AJAX, JavaScript / jQuery

Flask

* Backend development (REST API)

**Resources required**

Development Tools

IDE

* Development environments for coding

Libraries / Frameworks

Flask

* Developing web application, implementing security features

Python-docx

* Work with word documents in python

Xml parsing

* XML element tree library for parsing

Sample documents

* Word documents to validate application

Version control & Documentation

Git & Github

* To manage documentation, code collaboration and changes

**Team Members' Skills Distribution**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Skill Area** | **Current Capability (1 = Low, 2 = Medium, 3 = High)** | | | | | | | |
| **Elijah** | **Lok** | **Louis** | **Aidan** | **Fin** | **Jack** | **AVG** | **Value** |
| **Python Programming** | **3** | **2** | **2** | **2** | **2** | **2** | **2.17** | **Medium** |
| **Flask Backend** | **3** | **2** | **2** | **2** | **2** | **3** | **2.33** | **Medium** |
| **Frontend (HTML, CSS, Jinja)** | **2** | **3** | **1** | **1** | **2** | **2** | **1.83** | **Medium** |
| **Data Analysis (RSID comparison)** | **2** | **2** | **2** | **2** | **2** | **2** | **2.00** | **Medium** |
| **XML Parsing** | **1** | **1** | **2** | **2** | **1** | **1** | **1.33** | **Low** |
| **File Handling** | **3** | **1** | **2** | **2** | **1** | **1** | **1.67** | **Medium** |
| **Security Implementation** | **1.5** | **2** | **1** | **2** | **2** | **2** | **1.75** | **Medium** |
| **Testing** | **1** | **3** | **1** | **1** | **1** | **1** | **1.33** | **Low** |

**Skill comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skill area** | **Required for project** | **Current capability** | **Gap** | **Comments** |
| Python programming | 2 | 2 | 0 | Team has solid foundation, sufficient for project requirements. Practice in flask and data analysis could further strengthen this |
| Flask Backend | 2 | 2 | 0 | Adequate skill, beneficial to implement best practices in Flask including error and security measures. |
| Frontend (HTML, Jinja, CSS) | 2 | 2 | 0 | Sufficient, Can learn through freeCodeCamp, MDN or Coursera. |
| Data analysis (RSID comparison) | 3 | 2 | -1 | There is a gap, this is needed to analyze the RSID codes. Can learn on DataCamp |
| XML parsing | 2 | 1 | -1 | Need improvement, Prioritize learning XML parsing for RSID extraction and familiarization with RSID framework. Can learn on W3Schools or geekforgeeks |
| File handling | 2 | 2 | 0 | Adequate for project, can learn on Coursera |
| Security implementation | 2 | 2 | 0 | Adequate for web security. Can learn on OWASP, Coursera, and Udemy. |
| Testing | 2 | 1 | -1 | Gap in testing abilities, prioritize learning unit testing, integration, and performance testing. Can learn on Udemy |

**Resources comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Resource** | **Required for project** | **Availability** | **Gap** | **Comments** |
| IDEs | Yes | Available | None | Majority of team uses VSCode, ensure everyone comfortable with IDE |
| Git | Yes | Available | None | Git and Github used for version control. Team already uses and understands branching, merging and pull requests |
| Flask framework | Yes | Available | None | Flask is available, team has already set up main files and routing. |
| XML libraries | Yes | Available | None | Xml.etree library used for parsing files. Make sure team comfortable |
| Sample documents | Yes | Available | None | Sample docx needed to test validity of application, client has provided with some and a gen.py was made to create sample documents |