Multi-Criteria Decision-Making Workshops

INTRODUCTION

This document outlines the Multi-Criteria Decision Making (MCDM) workshops conducted as part of the tire procurement dashboard project for Truck Support Venlo. The MCDM workshops were organized to evaluate and select the best technologies and frameworks for the project based on various criteria such as performance, scalability, ease of use, and community support.

WORKSHOP OBJECTIVES

The primary objectives of the MCDM workshops were:

1. To evaluate various web scraping tools and techniques.
2. To select the most suitable backend framework for data processing and API development.
3. To determine the best frontend technology for developing a user-friendly and responsive dashboard.
4. To ensure that all selected technologies and frameworks align with the project requirements and stakeholder expectations.

PARTICIPANTS

- Project manager

EVALUATION CRITERIA

The following criteria were used to evaluate the technologies and frameworks:

1. **Performance:** Speed and efficiency in executing tasks.
2. **Scalability:** Ability to handle increasing amounts of work or data.
3. **Ease of Use:** Simplicity in implementation and maintenance.
4. **Community Support:** Availability of resources, documentation, and support from the developer community.
5. **Cost:** Licensing fees, development costs, and maintenance expenses.
6. **Security:** Robustness in protecting data and ensuring secure operations.
7. **Compatibility:** Ability to integrate with existing systems and tools.

WORKSHOP STRUCTURE

The MCDM workshops were conducted in three main phases:

1. **Preparation Phase:**
    - Identifying the technologies and frameworks to be evaluated.
    - Defining the evaluation criteria and their respective weights based on project priorities.
    - Preparing evaluation templates and scoring sheets.
2. **Evaluation Phase:**
    - Presentations and demonstrations of each technology/framework by the respective experts.
    - Hands-on sessions where participants could experiment with the tools.

- Scoring each technology/framework against the defined criteria.

3. **Decision-Making Phase:**
   - Aggregating the scores from all participants.
   - Discussing the results and resolving any discrepancies.
   - Making final decisions on the selected technologies and frameworks.

TECHNOLOGIES AND FRAMEWORKS EVALUATED

1. **Web Scraping Tools:**
   - **Selenium:**
     - Strengths: Versatile, supports multiple browsers and languages, handles dynamic content.
     - Weaknesses: Resource-intensive, setup complexity.
     - Score: High on performance, community support; moderate on ease of use, scalability.
   - **BeautifulSoup:**
     - Strengths: Simple to use, efficient HTML parsing, integrates well with other Python libraries.
     - Weaknesses: Limited to static content, slower for large datasets.
     - Score: High on ease of use; moderate on performance, scalability.
   - **Scrapy:**
     - Strengths: High-performance, built-in features for large-scale scraping, strong community support.
     - Weaknesses: Steeper learning curve, potential overkill for simple tasks.
     - Score: High on performance, scalability, community support.
2. **Backend Frameworks:**
   - **Flask:**
     - Strengths: Lightweight, easy to learn, flexible, strong community support.
     - Weaknesses: Limited built-in features, can require additional libraries for larger projects.
     - Score: High on ease of use, community support; moderate on scalability, performance.
   - **Django:**
     - Strengths: Full-featured, includes authentication, ORM, admin interface.
     - Weaknesses: Can be complex for small projects, steeper learning curve.
     - Score: High on scalability, security, community support.
   - **Express.js:**
     - Strengths: Lightweight, fast, robust, flexible, great for RESTful APIs.
     - Weaknesses: Less opinionated, more boilerplate code needed.
     - Score: High on performance, scalability; moderate on ease of use.
3. **Frontend Technologies:**
   - **Flutter:**
     - Strengths: High-performance, rich UI, fast development cycle with hot reload.
     - Weaknesses: Still relatively new, less mature ecosystem.
     - Score: High on performance, ease of use; moderate on community support.
   - **React:**
     - Strengths: Strong community support, highly performant, reusable components.
     - Weaknesses: Can be complex to set up, steep learning curve for beginners.
     - Score: High on performance, community support, scalability.
   - **Vue.js:**
     - Strengths: Easy to learn, flexible, great documentation.
     - Weaknesses: Smaller community compared to React.
     - Score: High on ease of use, performance; moderate on scalability, community support.

RESULTS AND DECISIONS

Based on the scores and discussions during the workshops, the following technologies and frameworks were selected:

1. **Web Scraping Tool: Selenium**
    - **Rationale:** Selenium was chosen for its versatility and ability to handle dynamic content, which is essential for scraping data from modern websites.
2. **Backend Framework: Flask**
    - **Rationale:** Flask was selected due to its simplicity, flexibility, and strong community support, making it ideal for developing the backend API and data processing workflows.
3. **Frontend Technology: Flutter**
    - **Rationale:** Flutter was chosen for its high-performance, rich UI capabilities, and fast development cycle, ensuring a responsive and user-friendly dashboard.