1. Project Overview

1.1 Project Objectives

This project as for goal to build a car rental application that will be both efficient and well designed for customers and car rental agency. This web application will serve as a platform for customers to be able to browse available vehicles and book a reservation over a determined time frame. It will also allow customer service representative to use these platforms to manage these reservations from the check-in to check-out.

1.2 Scope

This project is addressed for vehicle rental agencies in densely populated urban centers, that can also be popular tourist destinations. The key features and functionalities for users of this app will include :

- Browse available vehicles with or without filters
- Book a reservation for a vehicle fully online
- Modify and/or cancel a reservation
- Find the nearest branch based on postal code
- Vehicle review system to improve experience for each customers

From the agency point-of-view, they will have the possibility to access the reservations made by clients online and by their customer service agents,they will also have the ability to manage the cars in the database and finally, they will have access to user accounts.

1.3 Target Audience

The targeted users needs to be in possession of a **valid driving license** and be at least **18 years old**. The targeted audience includes individuals that needs to :

- travel a long distance with an adapted vehicle for the road
- move around in a city while they are visiting
- move out of their apartment
- move furniture that do not fit a regular car and many more needs that require transit that could not be fulfilled by their current means of transportation.

2. Project Approach

2.1 Development Methodology

The method chosen for the development of the software is Agile. It aligns with our project requirements since it allows for iterative development that will be separated in 4 different sprints, allowing us to improve our project based on feedback. It also allow us a close-collaboration between the contributors due to the regular meetings and the planning of the sprints, making communication a lot easier. They represent the main reasons why we believe that this methodology will allow us to be more efficient.

2.2 Project Timeline

The estimated timeline and milestones for the project are given below in 4 different sprints across the semester.

Sprint 1:

- Set up the GitHub repository and the documentation
- Find the technological approach in terms of frameworks and programming language
- Prepare the user stories and break them in tasks
- Task assignments for the team and prepare meeting environment
- Planning for the next sprint

Sprint 2:

- TBD

Sprint 3:

- TBD

Sprint 4:

- TBD

2.3 Collaboration and Communication

The main channel of communication for the team is located in discord where we plan out meetings, discuss changes and improvements. The result of these meetings will be described and written in the wiki. We will also use the Projects tab, to assign issues to collaborators and we will also comment on every changes/commit that has been done in GitHub. This will allow to keep a tab on which individuals contributed to which issues.

3. Technology Stack

3.1 Backend Frameworks

3.1.1 Flask

- Description:
  - Flask is a back-end framework. It provides the programmer with tools, libraries, and technologies that allow him to build a web application. This web application can be some web pages, a blog, a wiki, or go as big as a web-based calendar application or a commercial website. Flask is part of the categories of the micro-framework. They are frameworks with little or no dependencies to external libraries making the framework lighter, but on the other hand, the programmer has to do more work by himself.

- Rationale:
    - As beginners in web development, we wanted to have a web application development framework that is straightforward to use and understand, and we found that Flask provides handy documentation and a supportive community, and a minimalistic core since it equips developers with the necessary components without the excess baggage that often plagues larger frameworks, which makes the codebase more comprehensive, and we've found that flask facilitates for the user to customize the code as per his requirements. scalability, and ease of integration.
- Qualitative Assessment:
    - Strengths:
        - Flask is renowned for its minimalism and simplicity. This makes it an excellent choice for microservices, quick development, and small to medium-sized projects. Flask gives developers the freedom to select the tools, libraries, and extensions that they want. The web application can be developed in any way they wish. Because of Flask's vibrant development community, there are a ton of tools, guides, and extensions available to assist the developer.
    - Weaknesses
        - Flask has a lack of Built-in Features. Flask's minimalism is a pro for some, but it can be a con for others who need more built-in features, which are available in other frameworks like Django. As Flask projects grow in complexity and size, Flask may require more manual configuration and effort to manage. Security and Authentication: Flask provides the tools to build secure applications, but it places a significant amount of responsibility on the developer to implement security features correctly.
    - Use cases
        - Flask builds various types of web applications, including blogs, e-commerce sites, social media platforms, and more. Its simplicity and flexibility make it suitable for both small projects and larger, more complex applications.

3.2 Frontend Frameworks

3.2.1 HTML

- Description:
    - HyperText Markup Language (HTML) is the set of markup symbols or codes inserted into a file intended for display on the Internet. The markup tells web browsers how to display a web page's words and images. Each individual piece markup code (which would fall between "<" and ">" characters) is referred to as an element, though many people also refer to it as a tag. Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

- Rationale:
    - We chose HTML because Its basic tag-based structure helps web designers create awesome and interactive web pages that can be used to attract customers and readers to the car rental website. Users tend to learn the concepts and implement tags faster and easier. If you compare with the features offered by HTML5, there are many features such as video, audio, and multimedia, that you can use and implement for creating awesome websites for anyone. They provide also interactivity that can be added to static web pages thereby making them dynamic.
- Qualitative Assessment:
    - Strengths:
        - Ease of use: HTML is easy to learn and understand, especially for beginners. Its syntax is straightforward and uses simple tags to define the structure and content of a web page. This makes it accessible to individuals who are new to coding or web development. Since HTML has a simple structure, regular practice enables coders to easily write the HTML code.
        - Compatibility with different web browsers and devices: HTML is a standard markup language that is supported by all major browsers, including Chrome, Firefox, Safari, and Internet Explorer. This ensures that web pages created using HTML can be accessed and displayed correctly across various platforms, such as desktop computers, laptops, tablets, and smartphones.
    - Weaknesses
        - Lengthy code: Even for simple web pages, a lot of code needs to be written. Also, lengthy code produces complexity for the development team. To overcome this issue, we can use templates for basic web pages.
        - Not suitable for complex programming tasks: HTML is not a programming language. It is a markup language used for defining the structure and presentation of web content. This means that HTML cannot perform calculations, handle data manipulation, or implement complex logic like programming languages such as JavaScript or Python. As a result, developers often need to use additional programming languages or frameworks to achieve more advanced functionality on their websites.
    - Use Cases
        - HTML has multiple use cases. It is used in web page development, where different websites we see on the internet are regularly written in some form of HTML code. Navigating between web pages, responsive designs, storage functions in the browser, data entry support, the creation of web documents, game development, and offline web applications.

3.2.2 Bootstrap

- Description:
  - Bootstrap is a free, open source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs. As a framework, Bootstrap includes the basics for responsive web development, so developers only need to insert the code into a pre-defined grid system. The Bootstrap framework is built on Hypertext Markup Language (HTML), cascading style sheets (CSS) and JavaScript. Web developers using Bootstrap can build websites much faster without spending time worrying about basic commands and functions.
- Rationale:
  - We chose Bootstrap since it enables designers and developers to build completely responsive websites quickly. Since we are bound to an extremely confined timeline of few months to build a web application or a mobile application, we can easily take advantage of the Bootstrap framework and nail our project effortlessly. It is due to the ready-made blocks which are built ready for us to use them. We do not have to start everything from scratch and can modify certain elements to make them unique with the inputs.
- Qualitative Assessment:
  - Strengths:
    - One of the most notable advantages of Bootstrap is its inclusion of major components. Essentials like dropdown menus, navigation bars, forms, and progress bars are pre-designed and ready for use. You don't need to build these elements from scratch; instead, you can simply drop them into your project, saving you valuable time and effort.
    - In terms of the underlying technology, Bootstrap leverages JavaScript and CSS, two of the most widely used programming and markup languages in the world.
    - Bootstrap boasts a thriving community. Countless developers and enthusiasts are constantly working to enhance Bootstrap and offer support to fellow users. This means you can find answers to your questions, access resources, and stay up-to-date with the latest developments within the Bootstrap ecosystem.
  - Weaknesses:
    - The first Cons is that Bootstrap assumes a basic knowledge of coding. If the developer is completely new to coding, Bootstrap might feel like a maze with no clear path.
    - Another challenge with Bootstrap is its naming scheme, which can be perplexing. To effectively utilize Bootstrap, the developer needs to invest time upfront to learn its components and classes.

        - Using Bootstrap involves extensive documentation reading because there's no inherent way to deduce what's available or how it's named. The learning curve might be steeper for newcomers to web development.

- Use cases:
    - Bootstrap enables designers and developers to build completely responsive websites quickly.

4. Integration

4.1 Backend-Frontend Integration

- For a given API endpoint that we want to produce an HTML output, E.g a proper user interface or a form, we can attach template files written in HTML to pose as the return value of the given url path such as home.html. To do so, we need to import the render_template dependency, so then we can return the user-created page depending on the API endpoint by specifying the HTML template source folder like template_folder="templates".

5. Security Considerations

1. Basic HTTP authentication:
    - Use the built-in HTTP authentication mechanisms provided by web servers
2. Password hashing:
    - Implement using libraries or frameworks that provide simple APIs for password hashing functions.
3. Token-based authentication:
    - Use libraries or middleware that handle token generation and validation
4. Session-based authentication:
    - Use the built-in support for session management, making it easier to implement session-based authentication.
5. JSON/Ajax Support:
    - Enable support for JSON data exchange and Ajax requests, which can be achieved through libraries or frameworks that handle JSON parsing and AJAX requests.
6. Use of 'social'/OAuth for authentication:
    - Integrate OAuth libraries or SDKs provided by these providers.

6. Conclusion

The project adopts the Agile methodology to make use of the iterative development strategy, allowing for continuous improvements based on feedback. The technology stack chosen includes Flask for backend development, while HTLM will be complemented by Bootstrap for the frontend part of the operation. Informal communication will be made through Discord while the results of these meetings will be displayed on GitHub. Security measures will ensure user data protection and the authentication integrity.