# Programming Language 2: Course Work Report

# Course work title

**Write the Title of your course work**  
Python Web Browser

# Team Members and their responsibilities

**Name1 and responsibilities:Name2 and responsibilities:Name3 and responsibilities:Name4 and responsibilities:Name5 and responsibilities:**  
1) PM (Project Manager) Ulugbekov Almazbek, 2) FD (Frontend Developer) Osmonov Ulugbek, 3) BD (Backend Developer) Sultanov Temirlan, 4) UI/UX (Designer) Talantbekov Tilek

# Methodology

We used the PyQt5 framework for designing the user interface of the web browser. The backend was developed using Python, and the project was divided into 6-week milestones for better tracking and delivery. Tools like Git were used for version control, and regular meetings were held to ensure effective collaboration between the team members.  
  
We used the PyQt5 framework for designing the user interface of the web browser. The backend was developed using Python, and the project was divided into 6-week milestones for better tracking and delivery. Tools like Git were used for version control, and regular meetings were held to ensure effective collaboration between the team members.

**Novelty of work compared with the existing works**  
The novelty of this project lies in its unique combination of features, design, and approach compared to existing web browsers and similar software applications

# Results and Discussions

**Discussion the results obtained in your work**  
The results obtained from this project indicate that the development of a basic web browser using Python and PyQt5 was successful in meeting the primary objectives, but several limitations and opportunities for improvement were identified during the development process

# Objective

**:**

* The primary objective of this coursework is to design and develop a functional web browser application using Python and the PyQt5 framework.
* Develop a Basic Web Browser: Create a lightweight, functional web browser that supports fundamental features like navigating to URLs, refreshing pages, and going back and forth between pages.
* Integrate Front-End and Back-End Development: Combine front-end development with PyQt5 for designing the user interface (UI) and back-end development using Python for handling URL requests, error management, and maintaining a browser history.
* Enhance UI/UX Design: Develop an intuitive and visually appealing interface that allows users to interact seamlessly with the browser. This includes designing the layout, implementing buttons for actions like back, forward, reload, and input fields for URL navigation
* Test Functionality and Responsiveness: Ensure the browser functions well across different screen sizes and works smoothly across various scenarios like loading different websites, handling errors, and displaying the history of visited sites.
* Learn Best Practices in UI/UX Design: Focus on making the browser visually appealing, with clear and simple design elements. Understand user needs and integrate feedback for continuous improvement

Real-Time Applicability

The web browser developed in this project has several real-time applications across different scenarios:

1. **Low-Resource Devices**:
   * **Application**: In environments with **older hardware** or **low-spec devices**, where mainstream browsers may not perform efficiently.
   * **Real-Time Use**: This lightweight browser can be deployed on devices with limited CPU and RAM, ensuring smooth performance and fast browsing.
2. **Embedded Systems**:
   * **Application**: Many embedded systems or **IoT devices** require a simple browsing tool without the complexity of heavy browsers.
   * **Real-Time Use**: This browser can be integrated into devices such as kiosks, smart TVs, or digital signage systems, providing basic browsing functions without requiring substantial resources.
3. **Educational Tools**:
   * **Application**: In **educational settings**, where the goal is to teach students about the fundamental aspects of web browsers and software development.
   * **Real-Time Use**: The modular structure and simplicity of the browser make it an excellent tool for teaching **Python programming**, **UI/UX design**, and **browser architecture**.
4. **Customized Web Browsing Solutions**:
   * **Application**: For businesses or industries that require a **custom web browsing experience** with specific features.
   * **Real-Time Use**: The browser’s modular nature allows organizations to tailor it for specific needs, such as creating an internal browser with custom interfaces, navigation restrictions, or specialized web content for certain environments.
5. **Resource-Constrained Environments**:
   * **Application**: In areas with limited internet bandwidth or in **remote locations** with minimal infrastructure.
   * **Real-Time Use**: The browser’s lightweight design allows it to function in environments where other browsers may struggle due to network limitations or hardware constraints.

# Conclusion

This project successfully developed a **basic web browser** using **Python** and **PyQt5**, featuring essential functions like URL navigation, back/forward buttons, and page refresh. The browser is lightweight, making it suitable for low-resource devices. It also includes effective **error handling**, providing clear messages for users.

While the project achieved its goals, areas for improvement include integrating a more advanced **rendering engine**, adding **security features**, and incorporating **advanced functionalities** like tab management.

This browser serves as a solid foundation for future enhancements and can be applied in **low-resource environments** or as an **educational tool** for learning browser development.