SHADOW WEB BROWSER

Guide: Head of Department, Professor Krishna Reddy K R

Bhaskar V B, Dinakara D N, Christopher S,Lakshmi Narayan S

Head of Departmen, professor Krishna Reddy K R, Computer Science and Engineering, SJM institute of Technology, Karnataka, India
Bhaskar V B, Computer Science and Engineering, SJM institute of Technology, Karnataka, India
Dinakara D N, Computer Science and Engineering, SJM institute of Technology, Karnataka, India
Christopher S, Computer Science and Engineering, SJM institute of Technology, Karnataka, India
Lakshmi Narayan S, Computer Science and Engineering, SJM institute of Technology, Karnataka, India

ABSTRACT

In the modern digital era, web browsers serve as the primary gateway to the internet, enabling users to access information, communication, and entertainment. However, most contemporary browsers like Google Chrome, Mozilla Firefox, and Safari are built on highly complex architectures with millions of lines of code, making them unsuitable for educational purposes or lightweight customization. This creates a significant gap for students, researchers, and developers who wish to understand browser internals or develop customizable browsing solutions. Furthermore, existing browsers are often platform-specific or rely heavily on closed-source rendering engines, limiting accessibility and flexibility for learning and experimentation.This project addresses these challenges by developing a cross-platform web browser using .NET MAUI (Multi- platform App UI), which enables deployment across Android with a single codebase. The proposed system features an innovative hybrid rendering architecture that combines native WebView controls for handling complex, dynamic web content with a custom-built lightweight HTML rendering engine designed to parse and display simplified web pages. This dual-rendering approach provides both real-world functionality and educational insight into browser mechanics, including DOM construction, rendering pipelines, and layout management. The application implements essential browser features such as navigation controls, bookmark management, browsing history, and tab management, while utilizing SQLite for persistent data storage and following MVVM architectural patterns for maintainable, scalable code. By bridging the gap between industrial-grade browser functionality and educational transparency, this project serves as both a functional browsing solution and a powerful learning tool for understanding web technologies and cross-platform application development

INTRODUCTION

In the modern digital era, access to the internet has become a fundamental need for communication, education, entertainment, and productivity. Web browsers play a pivotal role in enabling users to navigate and interact with the vast resources available online. From accessing web applications and cloud services to watching videos and reading news, browsers serve as the primary gateway to the World Wide Web. Most commonly used browsers today, such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge, are built upon sophisticated rendering engines and architectures that have evolved over decades. While these browsers offer robust performance, high compatibility, and a wide array of features, they are also immensely complex — often built on millions of lines of code. As a result, they are not suitable environments for learners or developers seeking to understand the inner workings of a browser or for those attempting to create lightweight, customizable browsing solutions. The core objective of this project is to develop a cross-platform web browser using .NET MAUI (Multi-platform App UI) — a modern framework by Microsoft designed for building applications with a single codebase across multiple platforms, including Android. The project not only aims to replicate basic browser functionalities but also introduces an innovative hybrid rendering model. This model combines the use of native WebView controls — which rely on underlying platform-specific rendering engines — with a custom- built lightweight HTML rendering engine designed to parse and display simplified web content. By combining real-world functionality with educational clarity, this project offers a powerful learning tool and a functional browser prototype. It allows exploration of essential topics such as rendering pipelines, DOM construction, cross-platform development, native APIs, and MVVM architecture

1. SYSTEM DESIGN

The system design phase translates the functional and non-functional requirements into technical architecture and workflows. For a web browser application, this includes how the system components interact, how data flows between modules, and how users engage with various functionalities. The design phase ensures clarity for development and acts as a blueprint to implement and test the application accurately.

1.1. HARDWARE REQUIREMENTS

Component	Minimum Requirement	Recommended Requirement
RAM	8 GB	16 GB or Higher
Storage	512 GB HDD or SDD	1 TB SSD
Display	13”1080p	15”+ Full HD or Higher
GPU	Integrated	Dedicated
Network	Stable broad band	>50 Mbps internet speed

1.2. SOFTWARE REQUIREMENTS

Category	Requirement
Operating System	Windows 11/mac OS
Framework	.NET MAUI (.NET 8 SDK)
IDE	Visual Studio 2022(with MAUI workloads
Emulators/Simulators	Android Emulator/iOS Simulator
Database	SQ Lite (local embedded database)
Markup Languages	HTML5
Languages Used	C#,XAML
Version Control	Git & GitHub
Others	NuGet Package Manager, MAUI Tool kit, Web View SDK

This section describes the essential hardware and software requirements necessary for the smooth development, testing, and execution of the proposed application. Proper system configuration ensures optimal performance, stability, and efficiency throughout the development lifecycle.
The **minimum hardware requirements** include **8 GB of RAM**, which is sufficient for basic development tasks and running lightweight emulators. However, for better multitasking and smoother performance, **16 GB RAM or higher is recommended**.

In terms of storage, a **minimum of 512 GB HDD or SSD** is required to store development tools, project files, and databases, while **1 TB SSD** is recommended for faster data access and improved system responsiveness. The display requirement begins at **13-inch 1080p resolution**, which allows comfortable coding and testing, whereas a **15-inch Full HD or higher display** is recommended for better visual clarity and productivity.

For graphics processing, an **integrated GPU** is sufficient for standard development needs, though a **dedicated GPU** is advised when working with advanced emulators or graphics-intensive applications. A **stable broadband internet connection** is necessary for basic operations such as downloading dependencies, while a speed of **more than 50 Mbps** is recommended for faster updates, cloud access, and collaborative development.

The **software requirements** include an **Operating System such as Windows 11 or macOS**, which supports all the necessary development tools and frameworks. The application is developed using the **.NET MAUI framework with the .NET 8 SDK**, which enables cross-platform application development with high performance and modern UI capabilities. The primary **Integrated Development Environment (IDE)** used is **Visual Studio 2022 with MAUI workloads**, which provides a powerful environment for coding, debugging, and testing.

For testing and simulation, **Android Emulator and iOS Simulator** are used to verify the application's behavior across multiple platforms. The application uses **SQLite** as its database, which is a lightweight, local embedded database suitable for efficient data storage and retrieval. The user interface is designed using **HTML5**, and the application logic is implemented using **C# and XAML**, which are core languages for .NET MAUI development. For source code management and collaboration, **Git and GitHub** are used as version control tools. Additional tools such as the **NuGet Package Manager, MAUI Toolkit, and Web View SDK** are utilized to manage dependencies, enhance UI components, and integrate web-based features.

Overall, these hardware and software requirements ensure reliable performance, effective development, and seamless execution of the application across supported platforms.

2. RESULTS AND DISCUSSION

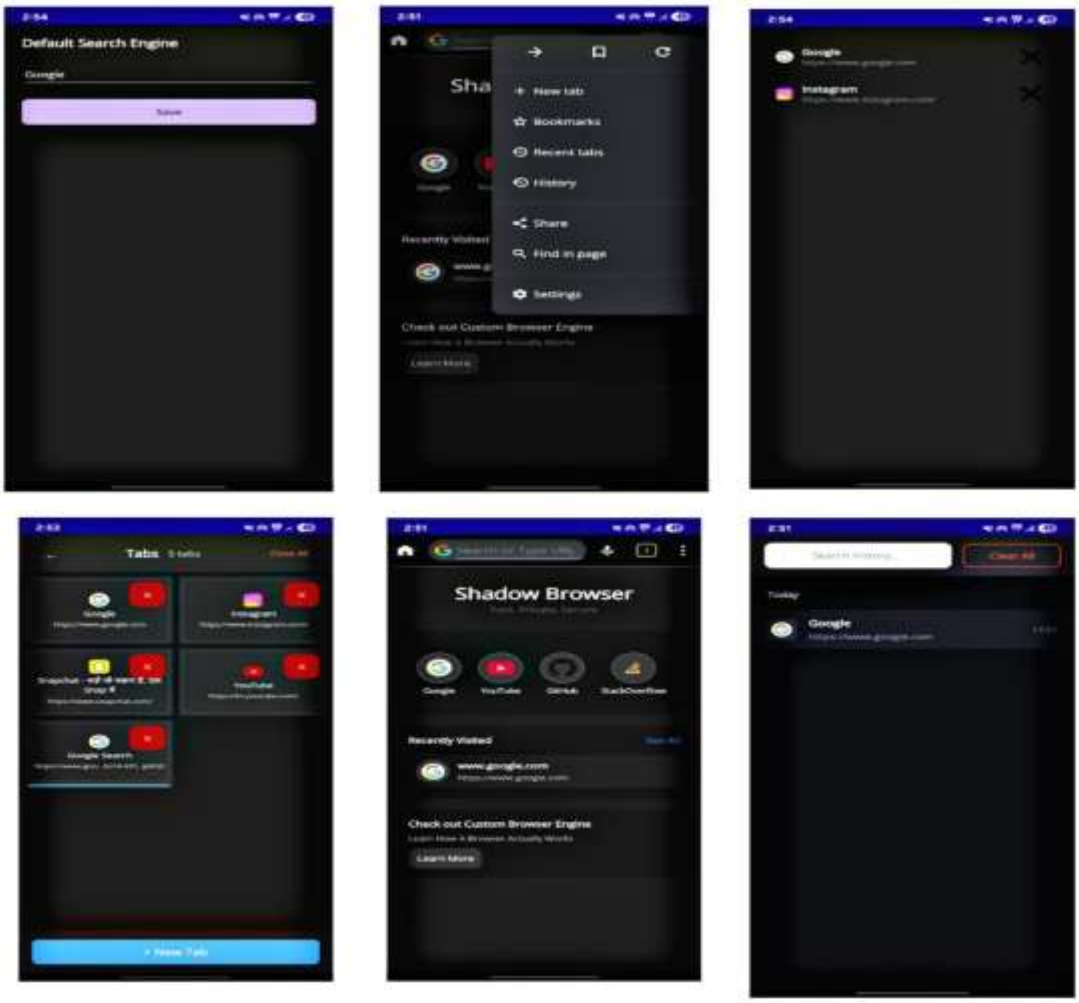

Fig – 1 Snapshots of our project

3. CONCLUSION

The successful development of the **Modern Cross-Platform Web Browser using .NET MAUI** demonstrates the effective integration of practical application development with educational objectives. This project achieves its primary goal of building a functional hybrid web browser that supports both **native WebView rendering** and a **custom-built HTML rendering engine**, offering users not only real-world usability but also valuable technical insight into browser internals.

By implementing essential browser features such as **URL navigation, back/forward navigation, refresh, bookmarks, history management, and persistent storage using SQLite**, the system provides a complete and user-friendly browsing experience. The use of **MVVM architecture** ensures clean separation of concerns, improving maintainability and scalability of the application. The adoption of **.NET MAUI** enables true cross-platform compatibility, allowing the same application to run seamlessly on **Android** with consistent performance and user interface.

From an academic perspective, the custom rendering engine plays a crucial role in simplifying complex browser mechanisms such as **HTML parsing, DOM creation, layout generation, and rendering**, making them accessible for learning and experimentation. This hybrid approach bridges the gap between industrial-grade browsers and educational tools.

In conclusion, this project successfully delivers a **scalable, educational, and functional web browser framework**. It serves as a strong foundation for future enhancements such as CSS extensions, JavaScript support, browser security features, and developer tools, making it a valuable contribution to both academic study and practical software development.

4. REFERENCES

[1] **IEEE Xplore: "A Reference Architecture for Web Browsers ".** Presents a layered reference architecture for web browsers, discussing the separation of concerns across networking, rendering, and user interaction modules. Useful for designing your browser's architecture using .NET MAUI. https://ieeexplore.ieee.org/abstract/document/1510168

[2] **Journal of Systems and Software (JSS): "A Reference Architecture for Web Browsers".** Explores web browser design patterns and compares implementations like Mozilla and Internet Explorer. Helpful for identifying reusable components and interaction models applicable to modern browsers. https://plg.uwaterloo.ca/~migod/papers/2006/jss-browserRefArch.pdf

[2] **ResearchGate / IEEE CAL: "Browser Bench: Towards a Benchmarking Framework for Web Browsers ".** Discusses techniques and tools for evaluating the performance of web browsers across metrics such as rendering speed, memory usage, and responsiveness. Useful when optimizing the performance of your .NET MAUI-based browsers. https://spark.ece.utexas.edu/pubs/CAL-22-browser-benchmark.pdf

[3] **About .NET MAUI:** Overview of .NET MAUI as Microsoft's cross-platform UI framework for building native Windows, Android, iOS, and macOS apps using C#. https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui

[4] **.NET MAUI App Architecture Guide:** Recommended structure for MVVM, dependency injection, and clean code practices in MAUI apps. https://learn.microsoft.com/en-us/dotnet/architecture/maui/overview

[5] **Http Client in .NET:** Standard .NET class for handling HTTP requests. https://learn.microsoft.com/en-us/dotnet/api/system.net.http.httpclient