ROOMCAST: MULTI-SCREEN MEDIA CASTING PLATFORM

Authors
Aleksandar Ivanov - S8090291
Nathan Goncalves - S8081735

Course
Course: NIT3004 - IT Capstone Project
University: Victoria University
Lecturer: Dr Gongqi Lin

References

- .Microsoft. (2025). ASP.NET Core MVC Documentation. Microsoft Learn. Retrieved from https://learn.microsoft.com/aspnet/core/mvc
- Microsoft. (2025). Entity Framework Core – Getting Started with SQL Server LocalDB. Microsoft Learn. Retrieved from https://learn.microsoft.com/ef/core
- Microsoft. (2025). Visual Studio 2022 IDE – Develop .NET Applications. Retrieved from https://visualstudio.microsoft.com/vs/
- SQLite Consortium. (2024). Comparison between SQLite and SQL Server LocalDB. SQLite Documentation. Retrieved from https://www.sqlite.org/

01 INTRODUCTION

RoomCast is a collaborative web-based application that enables users to upload, organize, and cast media content, such as document, images, and videos, onto multiple displays.
The primary idea of RoomCast is to enhance the ease with which users share and organize digital content in classrooms, workplaces, or spaces designed for collaborative work.
RoomCast runs on the ASP.NET Core MVC framework, the Entity Framework Core ORM, and the SQL Server LocalDB database to ensure a secure and reliable experience and performance. LibreOffice is used for accessible documentation viewing across all major operating systems. Users can connect to the RoomCast system via monitors, televisions, or remote receivers (e.g., Raspberry Pi) or lightweight client applications.

02 OBJECTIVE

The main aim of RoomCast is to make media management and screen casting easy for all users.
It allows users to log in, upload files, and display them on one or more connected screens.
The platform focuses on accessibility, cross-device compatibility, real-time media interaction and LibreOffice handles documentation conversions.
Future development will enable extended control using IoT-based clients.

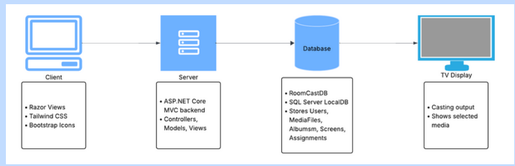

Figure 1: Flowchart of the client, Server, database and TV Display.

03 METHODOLOGY

RoomCast is designed with a client-server framework, operating from an ASP.NET Core MVC back-end structure querying a SQL Server LocalDB database. Users will interact with RoomCast through a clean browser interface where they can upload and organize their media.
The server handles the authentication, file processing and all communication with the database.
The stored media can then be viewed on any device connected to the client receiver (e.g., TV running a Raspberry Pi or Note++ client).
Testing and deployment was conducted using Visual Studio 2022 (v9.0.305 SDK). While Libre Office handles document conversion to pdf so user's can view the documents.

04 RESULTS

RoomCast offers good functionality in handling authentication, uploading files, creating albums, and assigning screens. Performance tests do indicate that the multimedia files loaded without lagging or difficulty. The application operates away from the need for users to establish a local database, to improve usability and effective set up time. RoomCast saves all data down securely via SQL Server LocalDB with guaranteed reliability across multiple sessions.

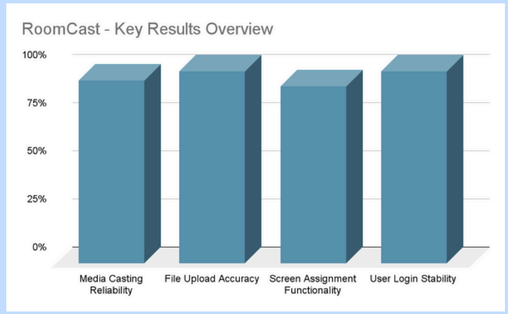

Figure 2: Bar chart of the success rate of each component in the system

05 DISCUSSION

RoomCast shows that a web-based casting system could operate properly, without external hardware like Chromecast or AirPlay.
This flexible construction will allow for future expansion into classrooms, offices, and collaborative workspaces.
Moreover, by using ASP.NET Core, it can quickly handle and authenticate data securely.

IMPORTANT!
The database (RoomCastDB) is pre-configured. Users do not need to run migrations or create tables manually. Simply open the project in Visual Studio 2022, build, and run — the system connects automatically.

06 CONCLUSION

RoomCast provides an efficient, user-friendly, and scalable way to manage and cast media across multiple devices. By combining simplicity with functionality, it reduces setup time and eliminates the need for external hardware. Future developments include expanding support for remote casting, improving video buffering, integrating cloud-based features for collaborative use.