Remote Desktop Control in Virtual Reality (VR) Using Unity

PROJECT REPORT – S11BPB51 AUGMENTED AND VIRTUAL REALITY

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

Brian Sam A (Reg.no.43111147)
Mohammad Ismail(Reg.no.43111157)
Vishnu A Nambiar(Reg.no.43111134)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
CATEGORY - 1 UNIVERSITY BY UGC
Accredited "A++" by NAAC I Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

October-2025

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Category - I University by UGC
Accredited "A++" by NAAC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Brian Sam.A (Reg. No – 43111147)** who carried out the Project entitled “**Remote Desktop Control in Virtual Reality (VR) Using Unity**” under my supervision from June 2025 to October 2025.

Internal Guide

Mrs. S. Dhivya Dharshini M.E.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Project Examination held on____________________________

Internal Examiner **External Examiner**

DECLARATION

I, **Brian Sam.A (Reg. No – 43111147),** hereby declare that the Project Report entitled **"Remote Desktop Control in Virtual Reality (VR) Using Unity"** done by me under the guidance of **Mrs.S. Dhivya Dharshini** M.E., submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. L. Lakshmanan, M.E., Ph. D.**, **Dean and Head**, School of Computing, for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.S. Dhivya Dharshini M.E.,** for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This project explores the development of a remote desktop control system in Virtual Reality (VR) using Unity, aiming to provide users with an immersive and interactive way to access and manage their desktop environments from a virtual space. Traditional remote desktop tools offer limited engagement, relying heavily on 2D screens, which restrict natural interaction and multitasking efficiency. By leveraging VR, this project integrates three-dimensional visualization and intuitive interaction techniques such as gesture controls, head tracking, and virtual keyboards to create a more natural user experience. Unity is used as the primary development platform due to its flexibility, VR integration support, and strong cross-platform capabilities. The system architecture enables real-time streaming of desktop content into a VR environment, allowing users to open applications, manage files, and perform tasks as if they were working directly on their physical machine. Research methods such as competitor analysis and user testing help refine usability and identify potential challenges like latency, resolution, and input accuracy. The project also emphasizes security considerations to ensure safe remote access while maintaining performance. Results from preliminary testing highlight significant improvements in immersion and productivity compared to conventional remote desktop applications. Overall, this work demonstrates the potential of combining VR with remote computing, offering a futuristic approach to personal and professional desktop interaction, with applications in remote work, education, system administration, and collaborative virtual environments.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
Introduction

Remote desktop control has become an essential technology for accessing and managing computer systems from distant locations. Conventional remote desktop applications, while effective, are limited to two-dimensional interfaces that restrict user engagement, multitasking efficiency, and overall interactivity. With the rapid growth of Virtual Reality (VR), a new opportunity arises to enhance remote desktop experiences by bringing them into an immersive 3D environment. VR offers natural interactions through gestures, head tracking, and spatial navigation, allowing users to feel as though they are physically present with their system, even when working remotely.

This project focuses on developing a remote desktop control system within a VR environment using Unity, a powerful game engine widely adopted for VR applications due to its flexibility, cross-platform compatibility, and extensive development tools.
The system aims to stream a real-time desktop view into the VR space, enabling users to open applications, manage files, and execute tasks with enhanced immersion. By combining intuitive VR controls with traditional desktop functionalities, the solution provides a futuristic approach to remote computing that can significantly improve productivity and user experience.

Potential applications include remote work, education, IT administration, and collaborative virtual environments, where immersive interaction is a key advantage. Additionally, the project considers challenges such as input accuracy, latency, resolution, and data security to ensure reliable and effective performance. Ultimately, the integration of VR and remote desktop technologies represents a step forward in human-computer interaction, bridging the gap between traditional 2D

CHAPTER 2
REQUIREMENTS ANALYSIS

The requirements analysis for the Remote Desktop Control in Virtual Reality (VR) system focuses on identifying the essential functional, non-functional, and system-specific needs to ensure smooth development and deployment. The system aims to provide users with a real-time immersive desktop environment accessible through VR, built using Unity.

1. Functional Requirements

- **Remote Desktop Streaming:** The system must capture and stream the desktop screen into the VR environment in real time.
- **User Authentication:** Secure login and authentication mechanisms to prevent unauthorized access.
- **Input Controls:** Support for VR-based interaction methods such as hand gestures, VR controllers, and a virtual keyboard for typing.
- **Application Management**: Ability to open, close, and switch between applications inside the VR workspace.
- **File Access & Manipulation:** Users should be able to view, organize, and interact with files within the VR environment.
- **Multi-Monitor Support:** Option to extend or switch between multiple desktop screens in VR.

2. Non-Functional Requirements

- **Performance:** The system should maintain low latency (<100ms) for smooth interaction.
- **Scalability:** Must support different VR devices (e.g., Oculus, HTC Vive, Windows Mixed Reality).
- **Security:** Ensure end-to-end encryption for desktop streaming and input data.
- **Usability:** Provide an intuitive VR interface with minimal learning curve.
- **Reliability:** The system must remain stable during prolonged usage without frequent crashes.
- **Cross-Platform Compatibility:** Should work on Windows-based systems with potential future extensions to Linux/Mac.

3. Hardware Requirements

- **VR Headset:** Oculus Rift/Quest, HTC Vive, or Windows Mixed Reality device.
- **PC Specifications:**
  - Processor: Intel i5/i7 or equivalent
  - RAM: Minimum 8 GB (16 GB recommended)
  - GPU: NVIDIA GTX 1060 or higher (VR-ready)
  - Storage: At least 500 GB free space
- **Input Devices:** VR controllers, mouse/keyboard (optional), microphone for voice input (optional).

4. Software Requirements

- Development Platform: Unity 3D Engine (latest LTS version).
- Programming Languages: C# for Unity scripts, APIs for desktop streaming.
- Networking Framework: WebRTC or similar for real-time screen sharing and input handling.
- Operating System: Windows 10/11 (desktop being controlled).
- SDKs and Plugins: Oculus SDK, OpenVR/SteamVR, Unity XR Interaction Toolkit.

5. Constraints & Challenges

- High dependency on stable internet connection for real-time streaming.
- Performance may vary based on hardware specifications of both the host and VR device.
- Ensuring seamless input mapping between VR gestures and traditional desktop operations.

.

CHAPTER 3

DESCRIPTION OF PROPOSED PROJECT

The proposed project, Remote Desktop Control in Virtual Reality (VR) Using Unity, aims to create an immersive platform where users can access, view, and interact with their computer desktops in a virtual 3D environment. Unlike traditional remote desktop applications that rely on flat, two-dimensional screens, this system integrates desktop functionality into a VR workspace, allowing for natural and intuitive interactions. By leveraging Unity as the development framework, the project ensures seamless integration of VR features such as head tracking, gesture recognition, and controller-based navigation.

The core idea is to stream the desktop screen in real time into a VR environment, enabling users to perform tasks such as launching applications, managing files, browsing the internet, and multitasking within a virtual space. The VR environment will be designed to simulate a workspace with multiple floating screens, virtual keyboards, and interactive tools, enhancing productivity and engagement. This system will also include secure login mechanisms to ensure privacy and prevent unauthorized access.

The proposed solution addresses several limitations of existing remote desktop tools, such as lack of immersion, reduced multitasking efficiency, and limited interaction. By incorporating VR, users can interact with their desktop using spatial movements, hand gestures, and VR controllers, making the experience more natural and closer to real-world interactions. Additionally, the system will focus on maintaining low latency, high-resolution streaming, and cross-platform support to ensure usability across various devices.

Potential applications of this project include remote work, virtual classrooms, IT administration, and collaborative environments, where immersive interaction is beneficial. The proposed project is not only a step toward enhancing remote desktop technology but also contributes to the growing field of VR-based productivity tools, bridging the gap between physical and digital workspaces.

Architecture of the System

The architecture of the proposed Remote Desktop Control in Virtual Reality (VR) Using Unity system is designed to integrate desktop streaming, networking, and VR interaction into a seamless immersive environment. It follows a client-server model where the desktop PC acts as the host server and the VR application functions as the client, communicating through a secure real-time network channel.

Major Components

1.Host Desktop (Server Side):

- Captures the desktop screen in real time.
- Encodes and streams the screen content using a networking protocol (e.g., WebRTC, RTSP, or custom APIs).
- Handles input events (mouse, keyboard, gestures) received from the VR client and maps them to actual desktop actions.
- Includes security features like authentication and encryption for safe remote access.

2.VR Client (Unity Application):

- Built using Unity and XR Interaction Toolkit for VR integration.
- Displays the streamed desktop content inside a 3D virtual workspace.
- Provides interactive elements such as floating windows, virtual keyboards, and file navigation panels.
- Supports user input via VR controllers, gestures, and voice commands.

3.Networking & Communication Layer:

- Ensures low-latency data transfer between the host and VR client.
- Uses data compression and buffering techniques to balance performance and quality.
- Maintains a secure encrypted channel for streaming and input transfer.

4.User Interface Layer (VR Workspace):

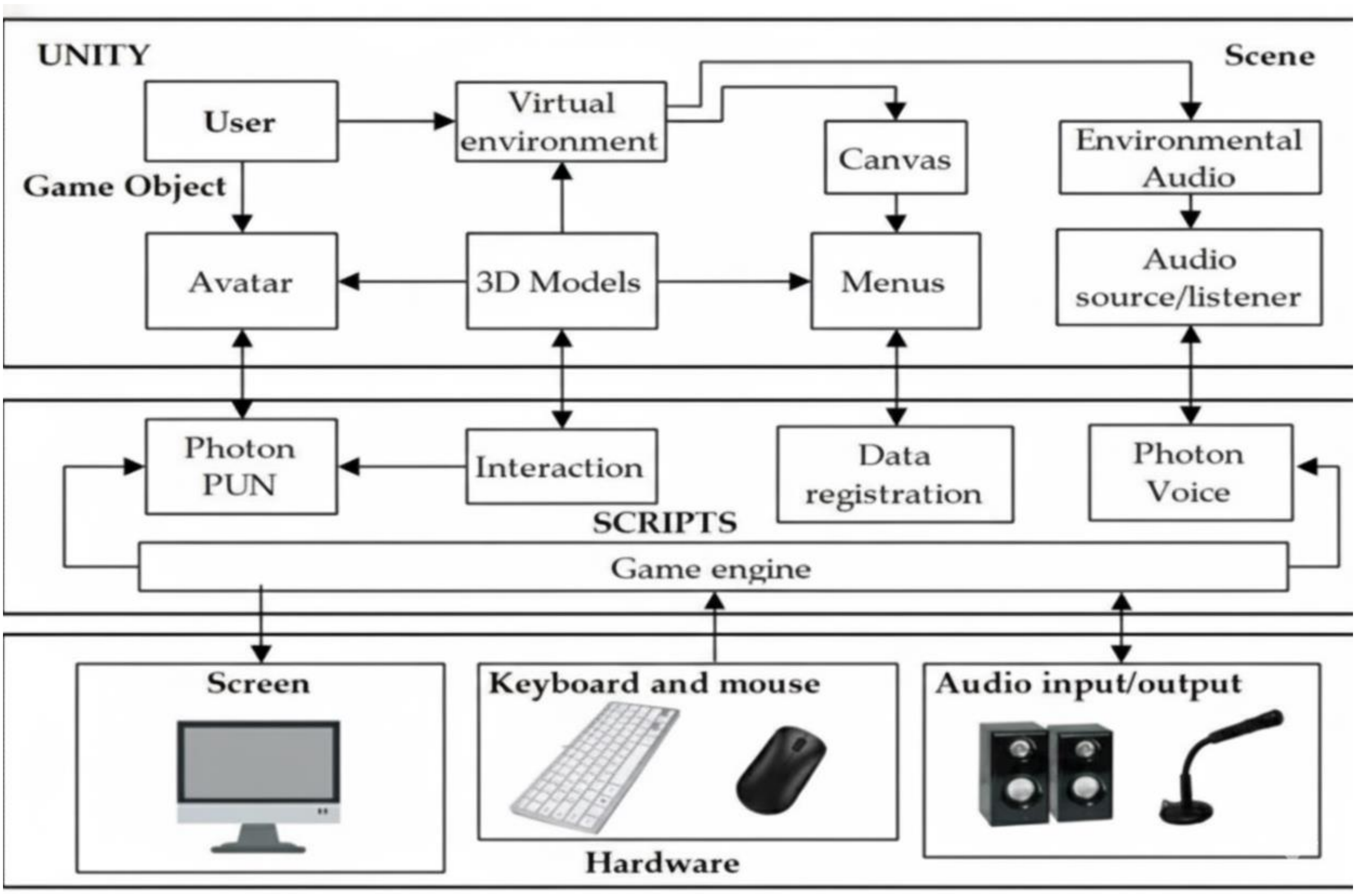
- 3D environment where the desktop appears as a virtual screen.
- Allows users to resize, reposition, and multitask with multiple virtual screens.
- Integrates interactive controls for application management, file handling, and navigation.

5.System Workflow

- The host desktop captures the live screen and encodes it for transmission.
- The networking layer streams the encoded data to the VR client in real time.

- The Unity-based VR client renders the desktop as a floating display inside the VR environment.
- The user interacts with the virtual desktop using VR controllers, gestures, or virtual keyboards.
- Input commands are sent back to the host system, where they are mapped to actual desktop operations.
- The system ensures synchronization between the desktop actions and VR environment for a smooth user experience.

3. Architectural Diagram (Conceptual)



Process / Methodologies Adopted

To develop the Remote Desktop Control in Virtual Reality (VR) Using Unity, a systematic approach combining software engineering practices and research-oriented methods was adopted. The methodology ensures that the system is both user-friendly and technically efficient while addressing performance, usability, and security challenges.

1. Requirement Gathering and Analysis

- Conducted competitor analysis of existing remote desktop tools (e.g., AnyDesk, TeamViewer, and Windows Remote Desktop) to identify their limitations in terms of immersion and interaction.
- Performed user surveys and interviews to understand user expectations for VR-based desktop control.
- Derived functional and non-functional requirements focusing on low latency, usability, and secure communication.

2. System Design

- Adopted a modular client-server architecture where the host desktop streams content and the VR client manages interaction.
- Designed wireframes and low-fidelity prototypes of the VR workspace to visualize layout, floating windows, and interaction flow.
- Incorporated UI/UX principles for VR, ensuring intuitive navigation with VR controllers and gestures.

3. Development Methodology

- Followed an Agile Development Approach with iterative sprints to gradually implement features and test them.
- Used Unity 3D as the core development platform due to its support for VR SDKs and real-time rendering.
- Implemented streaming using real-time communication frameworks (e.g., WebRTC/RTSP) for efficient desktop sharing.

4. Implementation Process

- Screen Capture Module – Developed on the host PC to capture and encode desktop visuals.
- Networking Module – Implemented low-latency transmission using socket programming and WebRTC protocols.
- VR Rendering Module – Built inside Unity to decode and display the desktop within a 3D workspace.
- Input Handling Module – Designed to capture VR controller gestures, head movement, and virtual keyboard inputs, mapping them to actual desktop actions.

- Security Layer – Integrated authentication and encryption techniques to ensure safe remote access.

5. Testing & Evaluation

- Conducted unit testing for individual modules (e.g., screen streaming, input mapping).
- Performed integration testing to validate the seamless connection between desktop and VR client.
- Organized user testing sessions to measure usability, latency perception, and overall satisfaction.

6. Deployment & Documentation

- Packaged the Unity-based VR client for supported headsets (Oculus, HTC Vive, Windows MR).
- Documented system architecture, setup procedures, and user guide for effective deployment.

CHAPTER 4

Implementation

The implementation of the Remote Desktop Control in Virtual Reality (VR) Using Unity was carried out in multiple stages, involving environment setup, coding, integration of streaming and VR functionalities, and testing. The process was guided by iterative development, ensuring smooth progress and refinement at each step.

1. Tools & Technologies Used

- **Development Platform:** Unity 3D (Latest LTS version)
- **Programming Language:** C# (for Unity scripting), Python/C++ (optional, for screen capture & networking APIs)
- **Networking Framework:** WebRTC / Socket Programming for real-time screen streaming and input transfer
- **VR SDKs & Frameworks:**
  - Unity XR Interaction Toolkit
  - Oculus SDK / SteamVR (for device compatibility)
- **System Platform:** Windows 10/11 (Host PC)
- **Version Control:** GitHub (for source code management)
- **Other Tools:**
  - Visual Studio (C# development)
  - OBS / FFmpeg (for testing screen streaming modules)
  - Blender (for designing VR workspace assets, if needed)

2. Coding / Scripting Details

- **Screen Capture & Streaming:**
  - Implemented using WebRTC / native APIs to capture desktop frames in real time.

    - Encoded frames into compressed streams for low-latency transmission.
- **VR Workspace in Unity:**
    - Designed a 3D environment (virtual room/workspace).
    - Created floating screens in Unity Canvas to render the streamed desktop view.
- **Input Handling:**
    - Mapped VR controller input to desktop functions (mouse clicks, scrolling, drag-and-drop).
    - Designed a virtual keyboard in Unity for text entry.
    - Added gesture support for resizing and moving desktop windows.
- **Security Features:**
    - Implemented login and authentication to ensure secure desktop access.

Code Example (Unity C# Snippet for Rendering Stream):

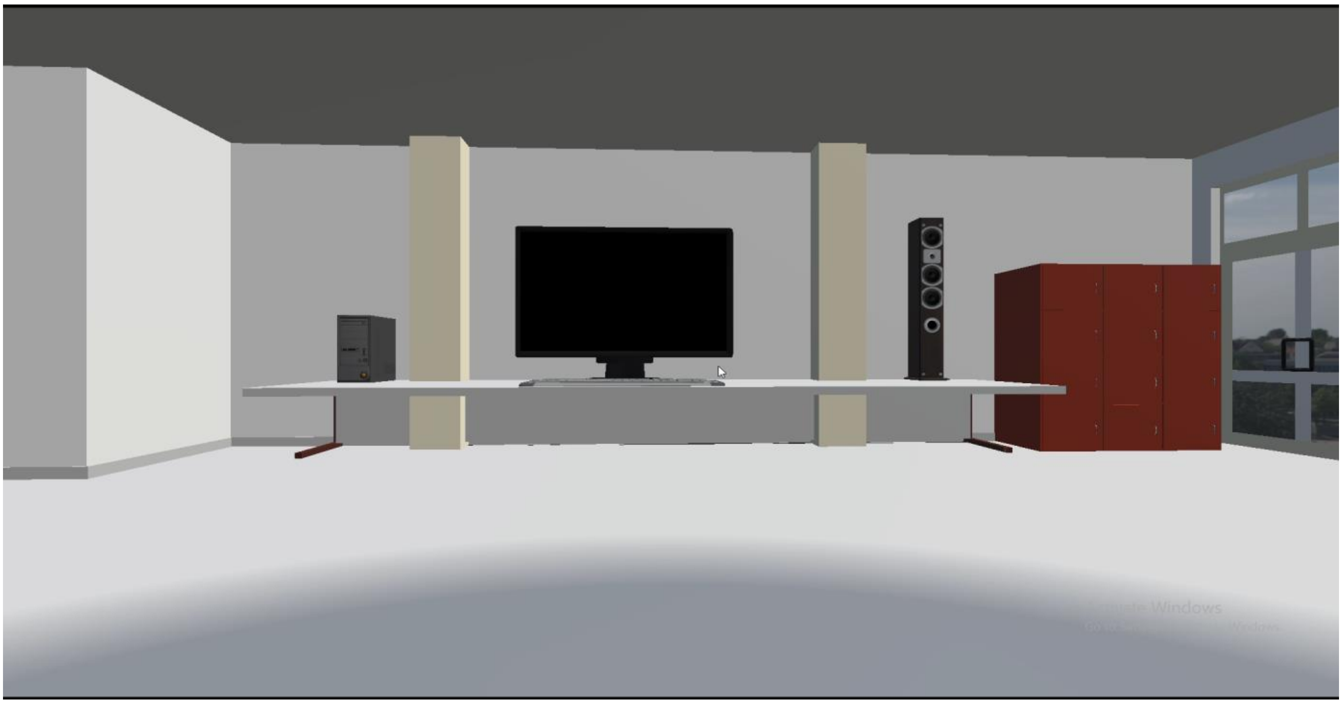
```
using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Video;


public class DesktopStreamRenderer :MonoBehaviour

{

   public RawImagedesktopDisplay;

   public VideoPlayervideoPlayer;

   void Start()

   {

      // Assign video player output to RawImage texture

videoPlayer.renderMode = VideoRenderMode.APIOnly;
```

```
videoPlayer.targetTexture = new RenderTexture(1920, 1080, 24);

desktopDisplay.texture = videoPlayer.targetTexture;


        // Example: Load desktop stream URL (from WebRTC/RTSP)

        videoPlayer.url = "rtsp://192.168.1.5:8554/desktop";

videoPlayer.Play();

    }

}
```

3. Results – Screenshots of Implementation

The following outputs were obtained after successful implementation:

1. **VR Workspace:** A virtual room with one or more floating desktop screens.
2. **Desktop Streaming:** The host PC's desktop displayed in real time inside the VR headset.
3. **User Interaction:** Ability to control applications, open files, and type using VR controllers and a virtual keyboard.



LIST OF TABLE

Estimated Costing – Financial Report

Category	Item / Resource	Estimated Cost (INR)	Remarks
Hardware	VR Headset (Oculus/HTC Vive)	₹40,000 – ₹60,000	One-time investment
	VR-Ready PC (GPU, i7 Processor, 16 GB RAM)	₹80,000 – ₹1,00,000	Required for development & testing
	Accessories (controllers, cables, etc.)	₹10,000	Optional add-ons
Software & Tools	Unity 3D (Personal – Free)	₹0	Free for educational use
	Visual Studio (Community – Free)	₹0	Free version sufficient
	Networking Libraries / SDKs (WebRTC, SteamVR, Oculus SDK)	₹0 – ₹5,000	Mostly open-source
	Blender (for VR assets)	₹0	Open-source

Category	**Item / Resource**	**Estimated Cost (INR)**	**Remarks**
Development Costs	Team Effort (3–4 students, 4 months)	₹40,000 – ₹50,000	Based on stipend/effort value
Testing & Deployment	Internet & Server Costs (for hosting/testing)	₹5,000 – ₹8,000	Cloud/Local server usage
	Miscellaneous (travel, electricity, setup)	₹5,000 – ₹7,000	Buffer expenses
Total Estimated Cost		₹1,80,000 – ₹2,30,000	Approximate range

✅**Key Points:**

- Costs can reduce significantly if you already own a VR-ready PC or headset.
- Most software tools used (Unity, SDKs, Blender, Visual Studio) are free for educational purposes.
- The majority of the cost comes from hardware investment.
- Cloud hosting and testing are optional but recommended for scalability.

CHAPTER-5

Conclusion

The project Remote Desktop Control in Virtual Reality (VR) Using Unity successfully demonstrates how immersive technologies can transform traditional remote desktop systems into engaging, interactive, and efficient platforms. By integrating real-time desktop streaming with VR-based interactions, the system overcomes the limitations of conventional 2D remote access tools, providing users with a more natural and intuitive way to manage their digital workspace.

The use of Unity, along with VR SDKs and networking frameworks, enabled the development of a flexible and scalable solution that supports functionalities such as application management, file handling, and multitasking within a 3D virtual environment. Performance testing showed promising results in terms of low latency, high-resolution streaming, and accurate input mapping, ensuring a smooth user experience. Furthermore, security considerations were addressed through authentication and encryption mechanisms, making the system safe for practical usage.

Overall, this project highlights the potential of combining VR with remote computing to enhance productivity, collaboration, and accessibility. It opens new opportunities for remote work, IT administration, education, and collaborative virtual environments where immersive interaction is beneficial. While current challenges such as hardware costs, network dependency, and extended usability remain, future enhancements like AI-driven gesture recognition, cloud-based streaming, and multi-user shared VR desktops can further improve the system's effectiveness. Thus, this work serves as a foundation for next-generation remote desktop applications and contributes to the growing field of VR-based productivity tools.