ANNEXURE: 1 MODULE WISE COURSE CONTENT AND OUTCOME

SL. NO	MODULE NAME	MODULE CONTENT	MODULE LEARNING OUTCOME	DURATION (HRS)
1	**Introduction to Unity**	**AR/VR Application Development in Unity** This competency focuses on designing and developing immersive AR/VR applications using Unity. It includes leveraging Unity's tools and features to create real-time, interactive virtual experiences for various industries.	- Master Unity's interface, scene management, and object hierarchies. - Develop, test, and deploy basic 3D/2D scenes using Unity GameObjects, prefabs, and assets. - Write and debug Unity scripts in C# to control object behaviors and interactions. - Build and export Unity applications for various platforms (PC, mobile, VR).	7.5 hrs
2	**Natural Language Processing**	**AI Model Integration in Unity for Real-Time Interactions** This skill competency emphasizes integrating AI models into Unity to enable dynamic interactions in real-time environments. It ensures seamless functionality between Unity objects and external AI tools like NLP or computer vision frameworks.	- Integrate NLP libraries like Dialog flow, Text to Speech or Hugging Face Transformers with Unity for text and speech processing capabilities. - Create Unity applications with real-time voice-to-text and text-to-speech functionality. - Develop conversational AI systems in Unity environments for immersive user interactions (e.g., voice-controlled avatars). - Combine NLP models with AR/VR features for	10 hrs

			real-time voice commands and interactive storytelling. - Implement Python-C# interoperability using Unity ML-Agents and external libraries like TensorFlow or PyTorch. - Deploy pre-trained AI models in Unity for tasks like object detection, speech processing, or text analysis. - Use Unity ML-Agents to create and train custom AI behaviors, such as decision-making agents in simulations. - Visualize AI models in Unity environments (e.g., heatmaps for object detection or bounding boxes for tracked objects).	
3	**Computer Vision Applications**	**Computer Vision Implementation for Object Detection and Tracking** This competency enables students to apply computer vision techniques in Unity to track and analyze visual data in real-time. It is critical for developing applications in	- Integrate OpenCV and TensorFlow models into Unity to enable real-time object detection and tracking. - Develop AR-based applications using Unity AR Foundation to overlay visual elements based on camera input. - Build gesture-controlled applications using hand-tracking technologies and computer vision	10 hrs

		robotics, AR navigation, and interactive gaming.	techniques. - Optimize Unity scenes for efficient processing of image recognition tasks on mobile or AR/VR devices.	
4	**AI Assistance**	**NLP Integration for Conversational and Interactive Interfaces** This competency develops the ability to build conversational AI systems within Unity, enabling natural language interactions. It is essential for applications in virtual assistants, customer service simulations, and interactive storytelling.	- Design and develop an AI-powered Unity project combining computer vision and NLP technologies, such as an AR chatbot or interactive virtual assistant. - Build applications for real-world scenarios, such as virtual tutors, smart surveillance systems, or AR/VR training modules. - Troubleshoot, debug, and optimize AI-integrated Unity projects for better performance and user experience. - Test applications on various platforms (desktop, mobile, VR) and ensure compatibility with deployment targets.	7.5 hrs
5	**End-to-End Application Development**	**End-to-End Project Execution for AI-Enhanced Applications** This skill emphasizes planning, developing, testing, and	- Implement advanced AR/VR interactions using Unity's XR Interaction Toolkit for immersive AI-based applications. - Develop multi-user interactive applications using Unity's networking and collaboration tools	10 hrs

		deploying comprehensive Unity-based projects that integrate AI for industry-specific applications. It involves managing all stages of development with a focus on achieving functional and performance benchmarks.	(Photon, Unity Netcode). - Create a complete AI-driven Unity application integrating computer vision, NLP, and AR/VR functionalities as a capstone project. - Present the final project with user documentation, demonstrating the workflow and technical skills acquired.	

ANNEXURE 2: Industry Use Cases/Final Projects

LEARNING OUTCOME	ASSESSMENT CRITERIA	PERFORMANCE CRITERIA	USECASES
Develop 3D Environments in Unity	Create functional 3D models and integrate them into interactive Unity scenes.	PC1: Create simple 3D objects (e.g., table, chair) using Unity tools. PC2: Apply basic textures and materials to 3D objects. PC3: Debug scene interactions.	**Virtual Room Designer**: Build a basic room setup with movable furniture using Unity's built-in 3D modeling and scripting tools. **Solution:** Build a Unity app where users can arrange virtual furniture in a 3D room. **Tools:** Unity (Personal Edition).
Build AI-Powered Chatbots	Design and integrate NLP-based conversational agents.	PC1: Set up an NLP model using ChatGPT API or Hugging Face. PC2: Script chatbot interactions for Unity-based environments. PC3: Customize responses for queries.	**Customer Support Bot**: Create a Unity app with a built-in chatbot that answers basic product-related FAQs using ChatGPT API. **Solution:** Embed a chatbot in Unity for text-based queries about products. **Tools:** ChatGPT API, Unity.
Implement Object Detection with Python	Develop basic object recognition systems using Python libraries like OpenCV.	PC1: Capture live video feed and apply object detection. PC2: Identify and label objects in real-time. PC3: Integrate detection output with Unity.	**AR Object Finder:** Create an app where the camera identifies basic objects (e.g., a water bottle) and displays labels using Unity and OpenCV integration. **Solution:** Use Python’s OpenCV for object recognition and display detected items in Unity’s AR interface. **Tools:** Python, OpenCV, Unity.

Create Immersive AR/VR Experiences	Design interactive AR/VR prototypes for real-world applications.	PC1: Use Unity XR Toolkit to build an AR/VR environment. PC2: Incorporate basic user interactions (e.g., button press). PC3: Optimize for smooth performance.	**VR Art Gallery:** Build a virtual space where users can view paintings and interact with art pieces (zoom, rotate) using Unity's XR tools. **Solution:** A Unity-based VR experience where users explore a virtual art gallery. **Tools:** Unity XR Toolkit.
Visualize Data with AI Models	Develop data-driven visualizations in Unity using AI-generated analytics.	PC1: Train a basic AI model in Python for data analysis. PC2: Generate insights (e.g., charts, summaries). PC3: Display insights in Unity.	**Interactive Dashboard:** Create an AR dashboard that displays sales performance, pulling analytics from Python models into Unity visuals. **Solution:** Develop an AR dashboard displaying live data insights powered by Python AI models. **Tools:** Python (NumPy, Matplotlib), Unity.

LIST OF FINAL PROJECTS	
SL.NO	**FINAL PROJECT**
1	*Virtual Classroom Setup:* *Students lack engaging virtual environments for learning. Solution: Build a basic virtual classroom with AI-powered avatars that interact and answer simple questions, providing an engaging and interactive learning space.*
2	**Object Sorting for Waste Management:** Waste segregation at source is often manual and inefficient. Solution: Develop a Unity simulation for object detection and classification, demonstrating how AI can improve waste sorting processes.
3	**Basic Conversational AI for FAQ Support:** Users struggle to find information on websites. Solution: Create a Unity-based chatbot system that answers FAQs, making it easier for users to access key information interactively.

4	**Virtual Tour for Local Park:** Visitors can't explore parks remotely or learn about key attractions. Solution: Build a simple VR tour of a park with AI-generated voice guides to enhance user experience and accessibility.
5	**Recipe Assistant App with AR Features:** People often don't know what to cook with available ingredients. Solution: Design an AR app where users scan ingredients to see recipe suggestions, making meal planning more efficient and fun.
6	**AR Virtual Tutor for Science Lab** Create an AR application where students can visualize and interact with virtual lab experiments guided by a voice-driven tutor.
7	**Smart Factory Maintenance Simulator** Develop a VR training simulator for factory workers to identify, diagnose, and fix machinery issues.
8	**AI-Driven Virtual Shopping Assistant** Design an AR-powered assistant that recognizes products in a store and provides voice-based recommendations based on customer preferences.
9	**VR Escape Room with AI Characters** Create a virtual reality escape room where players solve puzzles guided by AI characters that respond to voice commands.
10	**Driver Assistance VR Training System** Develop a training system for drivers to practice handling scenarios like obstacle avoidance and lane discipline in a simulated environment.
11	**Warehouse Inventory Management AR App** Build an AR application that automates inventory checks using object detection and voice commands for inventory tracking and updates.
12	**Medical AR Surgical Guide** Create an AR surgical guide that overlays real-time procedural steps on a mannequin or dummy for training purposes.
13	**Virtual Museum with AI Tour Guide** Develop an immersive virtual museum where visitors interact with exhibits through AI-powered conversational guides.
14	**AI-Based Traffic Control Simulation** Build a simulation for traffic management using AI to control vehicle flow and reduce congestion in real-time scenarios.
15	**Cable Car Troubleshooting Trainer** Create a VR-based training platform for cable car maintenance, focusing on troubleshooting mechanical and electrical faults.

ANNEXURE 3 – COURSE ASSESSMENT

COURSE ASSESSMENT RUBRICS (TOTAL MARKS: 70)				
ASSESSMENT CRITERIA	**DESCRIBE THE CRITERIA OF THE BELOW CATEGORY PERFORMANCE**			**TOTAL MARKS**
	FAIR	**GOOD**	**EXCELLENT**	
1 Functionality	Some features work, but there are noticeable bugs that hinder usability. The application has performance issues such as lag, crashes, or slow response times. Partially meets objectives; major aspects are missing or underdeveloped.	Most features work, with minor issues that do not affect usability. The application performs well with occasional delays or minor performance issues. Meets most objectives but misses minor aspects.	All intended features work flawlessly, with no bugs. The application performs smoothly with optimal response times across devices. Fully meets or exceeds project goals as outlined in the proposal.	20
2 Innovation	Relies on conventional approaches with little creativity. The solution is abstract or impractical for real-world scenarios.	Presents a somewhat innovative approach with some originality. Practical and useful but with limited real-world application potential.	Introduces a highly unique and creative solution to the problem, setting it apart from existing solutions. Highly applicable and realistic in a real-world context, solving a well-defined problem effectively.	15
3 Technical Implementation	Limited tool integration; some tools are used ineffectively	Successfully integrates most tools, with minor gaps.	Seamlessly integrates multiple tools (e.g., Unity, OpenCV, NLP libraries) to achieve	15

	. Code has significant issues such as poor structure, lack of documentation, or inefficiencies. Implements basic techniques with minimal exploration of advanced capabilities.	Code is functional and reasonably well-documented but contains minor inefficiencies. Uses moderately advanced techniques with reasonable accuracy.	objectives. Code is modular, well-documented, and error-free. Implements advanced techniques (e.g., ML models, AR/VR interactions) with high precision.	
4 User Experience	Basic design; some elements are unintuitive or lack visual coherence. Limited accessibility; some users may struggle with controls or navigation.	Clear design with some minor usability or aesthetic issues. Moderately accessible with minor usability concerns for some users.	Intuitive, visually appealing, and aligns with user expectations. Highly accessible to a diverse audience, with well-thought-out controls and features.	15
5 Presentation	The presentation is somewhat unclear or disorganized. Limited or incomplete supporting materials, with minimal effort evident.	The presentation is clear but lacks energy or engagement. Includes adequate slides and/or documentation but lacks polish or depth.	The presentation is confident, engaging, and clearly explains all aspects of the project. Includes polished slides, demo videos, and detailed documentation that enhance understanding.	5