

CAPSTONE PROJECT

SKILLITH

AGENTIC AI FOR PERSONALIZED COURSE PATHWAYS.

"YOUR AI COACH FOR SMARTER LEARNING DECISIONS"

Presented By:

Sai Videsh – IIITDM Kurnool – BTech_CSE

OUTLINE

- **Problem Statement**
- **Proposed Solution**
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT

Students often struggle to identify the right learning path that aligns with their interests and long-term goals due to the overwhelming number of online courses and a lack of personalized guidance.

LearnMate aims to solve this by acting as an Agentic AI coach that interacts with students, understands their interests (like Frontend Development, Cybersecurity, UI/UX Design, etc.), assesses their current skill level, and dynamically builds a personalized course roadmap that adapts over time based on progress and preferences.

PROPOSED SOLUTION

The proposed system aims to solve the issue students face in choosing the right learning path among the vast number of online resources. Learners often feel overwhelmed when trying to align courses with their personal interests, skill levels, and long-term goals. This solution introduces an AI-powered agent that delivers personalized, structured learning roadmaps based on user input. The agent adjusts and evolves the learning path dynamically over time based on user progress and feedback.

User Data Collection

- The system begins by collecting three core pieces of input from the student: their domain of interest (e.g., frontend development, cybersecurity), current skill level (beginner, intermediate, advanced), and a time-bound or career-specific goal (e.g., job-ready in 3 months).
- These inputs are used to generate a unique learning roadmap tailored to the individual.
- Optionally, the user can update their progress or resubmit information as their learning evolves, allowing the roadmap to adapt accordingly.

Data Processing and Personalization Logic

- Structured prompts are designed within IBM Watsonx Prompt Lab to map inputs to a curated 10- to 12-week learning plan.
- Each plan includes a week-by-week breakdown of topics, tools, hands-on tasks, and recommended certifications.
- The tone of the roadmap is intentionally practical, aiming to make the learner feel guided rather than overwhelmed.

PROPOSED SOLUTION

AI Model and Intelligent Behavior

- The core engine of the system is powered by IBM Granite, accessed through Watsonx Prompt Lab.
- The model functions as an agentic assistant, capable of not just responding to single prompts but tracking previous conversations to offer updated roadmaps based on new inputs.
- The agent can also handle follow-ups such as “I’ve completed week 3, what’s next?” and adjust the path accordingly.
- There is an option to integrate RAG (Retrieval-Augmented Generation) to pull real-time data or course listings for enhanced personalization.

Deployment and Platform Integration

- The front-end chat interface is implemented using IBM Watsonx Assistant or directly through the Prompt Lab interface.
- Backend logic and API routing (if required) are handled via IBM Cloud Functions or a lightweight Python/Flask application.
- The system is hosted entirely on IBM Cloud Lite, meeting the platform and service requirements for project deployment.
- Efforts have been made to ensure the system is scalable and accessible through a responsive and simple interface.

Evaluation and Feedback Loop

- The effectiveness of the roadmap is evaluated based on direct user testing.
- Feedback is collected through manual surveys or thumbs up/down functionality.
- Further improvements to the prompts and flow are made based on real-world usage.
- Success metrics include roadmap completion rate, session retention, and positive feedback ratio.

Result

The final solution, named Skillith, delivers personalized and adaptive learning plans to students using IBM’s LLM infrastructure and Cloud Lite services. The system provides meaningful, goal-driven guidance to help learners progress effectively toward their desired outcomes, whether job readiness, freelancing capability, or domain mastery.

SYSTEM APPROACH

■ 1. System Requirements

- List all the hardware, software, and platform-related requirements needed to build and test the solution. For example:
- A system with a stable internet connection
- Modern web browser (Chrome, Edge)
- IBM Cloud account (Lite tier or above)
- IBM Watsonx Studio and Runtime access
- Optional: IBM Cloud Functions (for backend hosting)

■ 2. Libraries and Tools Used

- This subsection lists all the necessary libraries, platforms, and tools:
- IBM Watsonx Studio
- IBM Watsonx Agent Builder (Agentic Lab)
- IBM Granite (LLM model)
- Mistral (for comparison testing)
- RAG file upload capability
- Optional: Python (Flask backend), IBM Cloud CLI, VS Code

SYSTEM APPROACH

3. Methodology and Development Steps

- Logged into IBM Cloud and deleted all existing resources for a clean environment
- Accessed Watsonx Studio and created new runtime and studio environments
- Opened Watsonx Agent Lab and followed the documentation to create a new agent
- Associated required services and generated an API key for access
- Provided a system prompt to guide the agent's behavior for generating learning roadmaps
- Uploaded a RAG document to act as a reference database for the agent
- Tested agent output across different LLMs including IBM Granite and Mistral
- Analyzed model responses and performance for accuracy and quality
- Deployed the final version once consistent results were obtained
- Due to time constraints, the UI and backend hosting were not implemented
- The project remains ready for future upgrades with a frontend and backend layer on IBM Cloud

ALGORITHM & DEPLOYMENT

This document provides an in-depth description of the system architecture, implementation steps, model configuration, and deployment process for the Agentic AI project, an AI-powered personalized learning coach. This system was designed to assist students in identifying the best course pathway based on their interests, skill levels, and goals. Built using IBM Watsonx and IBM Granite models, the solution fulfills both the practical need for personalization and the technical requirement of utilizing IBM Cloud Lite services.

1. Introduction to the Project Goal

- The Agentic AI system was conceptualized as a solution to address the overwhelming landscape of online learning platforms and courses, where students struggle to find direction aligned with their unique aspirations. The objective was to create an intelligent assistant capable of dynamically constructing and adapting personalized course roadmaps. This assistant, named Skillith, was designed to:
- Interact with users via natural language
- Collect three key inputs: interest area, current skill level, and time/career goal
- Generate clear, actionable, and motivational learning roadmaps (10–12 weeks)
- Use Retrieval-Augmented Generation (RAG) to enhance factual accuracy
- Provide guidance with real-world tools, mini projects, and optional certifications
- This solution was implemented using IBM Watsonx Studio, integrated with IBM Granite 7B models and deployed within IBM Cloud Lite infrastructure.

2. IBM Cloud Setup and Service Provisioning

- The implementation began by logging into the IBM Cloud portal. To avoid conflicts or resource limitations, all existing or unnecessary resources were deleted to free up space under the Lite plan, ensuring a clean slate.
- Using the IBM Cloud console, Watsonx Studio was searched and opened. A new Watsonx Studio project was created, named Skillith_Project. During creation, it was linked to a newly created instance of Watsonx Runtime, which is required for executing and managing AI models. This project serves as the central workspace for building and training models, integrating data, and managing AI agents.
- The Watsonx documentation for Agentic AI Labs was consulted to understand the structure and behavior of agent-based AI systems. Important details included how agents process user instructions, integrate retrieval-based knowledge using RAG, and utilize system prompts to define consistent personality and behavior.

ALGORITHM & DEPLOYMENT

3. Agent Construction Process

- Within Watsonx Studio, the "Build AI Agents" section was opened. A new agent named Skillith was created, linked to the active project and runtime instance. The agent was configured to allow external data retrieval and a new API key was generated to enable future integration or API access.
- A system prompt was added to the agent to define its behavior. This prompt explained that Skillith was a learning coach that gathers three inputs from the user (interest area, skill level, and time/career goal), and then generates a week-by-week roadmap in a simple, actionable format. The tone was set to be encouraging and beginner-friendly.
- To improve the factual accuracy of answers, a RAG knowledge file (plain text) was uploaded. This file contained curated information about learning paths, tools, mini projects, certifications, and more. The agent was configured to reference this RAG file when generating responses, allowing it to behave more like a domain-specific tutor.

4. Model Selection and Configuration

- Several models were available for powering the agent, including Mistral, LLaMA, and IBM Granite. After trying different models, IBM Granite 7B was selected due to its ability to understand intent well, structure responses consistently, and refer correctly to the RAG file.
- The final configuration settings used were:
 - Model: Granite-7B
 - Max Tokens: 2000
 - Temperature: default
 - Frequency Penalty: default
- This configuration was sufficient to allow multi-turn conversations, progress tracking, and switch in learning tracks (e.g. from Frontend to Cybersecurity) without breaking the conversation flow.

ALGORITHM & DEPLOYMENT

5. Testing and Iteration

The agent was tested with several user queries, such as:

- “I’m a beginner in Frontend Development. Want to be job-ready in 3 months.”
- “I’ve completed till Week 3. What’s next?”
- “I want to start freelancing in UI/UX Design within 6 weeks.”

Each time, the agent returned:

- A clear roadmap split by week
- Key tools to learn such as VS Code, Figma, GitHub, etc.
- Mini projects like designing landing pages, writing Python scripts, or training ML models
- Platform suggestions for learning (Coursera, freeCodeCamp, etc.)
- Optional certifications and follow-up guidance

Multiple rounds of testing were done to polish language, remove grammar mistakes, and refine the prompt further to make it more generalizable.

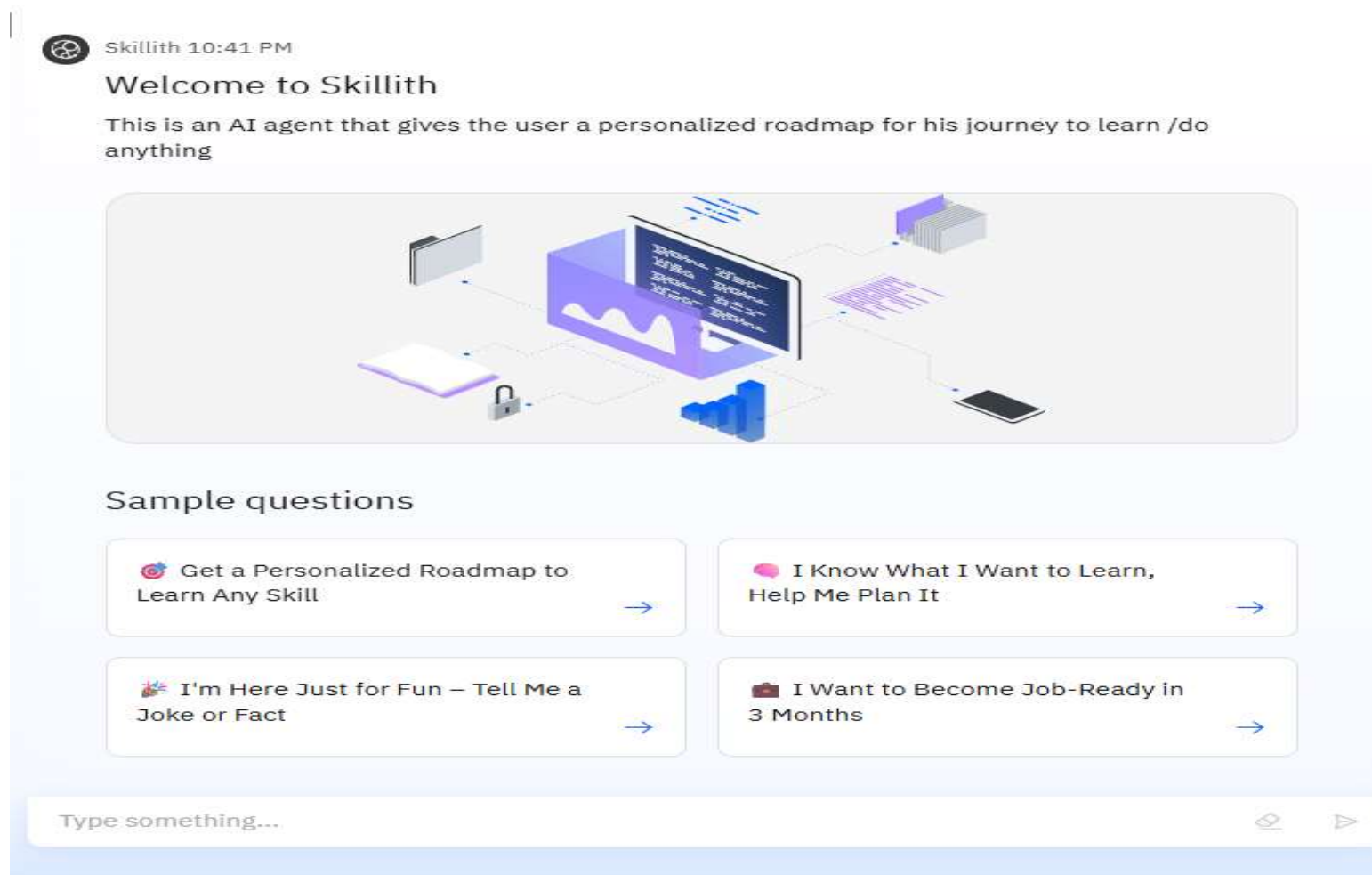
6. Deployment

Once the agent behavior was fine-tuned, it was deployed from Watsonx Studio itself. The deployment used the Agentic AI Deployment space named Agentic_AI_01.

The deployed version retained access to the same RAG database, model configuration, and system prompt. After deployment, the agent was again tested to ensure behavior matched the build version.

This deployed agent is now available for use via API or direct integration into a front-end.

RESULT



RESULT

Sure! To get started, I need to know a few things:

1. **Interest Area:** What specific skill or topic are you interested in learning? (e.g., AI/ML, Frontend Dev, Cybersecurity)
2. **Current Skill Level:** What is your current skill level in this area? (Beginner / Intermediate / Advanced)
3. **Career Goal or Time Goal:** What is your goal? (e.g., "Job-ready in 3 months")

Once I have this information, I can generate a personalized roadmap for you!

► How did I get this answer?

You: 10:49 PM

Block chain, beginner, I have only one month to learn and become a block chain developer

Skills: 10:49 PM

Great! Here's a personalized roadmap to help you become a Blockchain Developer in one month. This plan assumes you can dedicate 1-2 hours per day.

Week 1: Foundations of Blockchain

Topics to Learn:

- Introduction to Blockchain
- How Blockchain Works
- Cryptography Basics
- Consensus Mechanisms (Proof of Work, Proof of Stake)

Week 1: Foundations of Blockchain

Topics to Learn:

- Introduction to Blockchain
- How Blockchain Works
- Cryptography Basics
- Consensus Mechanisms (Proof of Work, Proof of Stake)

Tools/Platforms:

- Coursera: "Blockchain Specialization" by University of California, Irvine
- YouTube: "Blockchain Explained" by Simply Explained

Mini-Project:

- Create a simple blockchain in Python

Learning Outcome:

- Understand the basics of blockchain technology
- Implement a basic blockchain from scratch

Optional Certifications:

- Coursera Certificate for "Blockchain Specialization"

Week 2: Smart Contracts and Ethereum

Topics to Learn:

- Introduction to Ethereum
- Smart Contracts Basics
- Solidity Programming Language
- Writing and Deploying Smart Contracts

Tools/Platforms:

- Ethereum.org: Official Ethereum Documentation
- Remix IDE: Online Solidity Compiler

Mini-Project:

- Write and deploy a simple smart contract on the Ethereum testnet

Learning Outcome:

- Understand Ethereum and smart contracts
- Write and deploy smart contracts using Solidity

Optional Certifications:

- Certified Ethereum Developer (CED) by Blockchain Council

Week 3: Advanced Blockchain Concepts

Topics to Learn:

- Decentralized Applications (DApps)
- Token Standards (ERC-20, ERC-721)
- Blockchain Interoperability
- Scalability Solutions (Layer 2, Sharding)

Tools/Platforms:

- Udemy: "Ethereum and Solidity: The Complete Developer's Guide"
- Medium: Blockchain Articles and Tutorials

Mini-Project:

- Create a simple DApp using React and Ethereum

Learning Outcome:

- Understand advanced blockchain concepts
- Build a basic DApp

Optional Certifications:

- Certified Blockchain Developer (CBD) by Blockchain Council

Week 4: Real-World Projects and Job Preparation

Topics to Learn:

- Blockchain Use Cases
- Industry Trends and Job Market
- Resume and Portfolio Building
- Interview Preparation

Tools/Platforms:

- LinkedIn: Networking and Job Search
- GitHub: Showcase Your Projects
- LeetCode: Practice Coding Interviews

Mini-Project:

- Contribute to an open-source blockchain project

Learning Outcome:

- Gain real-world experience
- Prepare for job interviews

Optional Certifications:

- Certified Blockchain Professional (CBP) by Blockchain Council

Additional Resources

- **Books:** "Mastering Blockchain" by Imran Bashir, "Blockchain Basics" by Daniel Drescher
- **Communities:** Reddit (r/ethereum, r/blockchain), Stack Overflow, Ethereum Dev Forums

Tips for Success

- Stay consistent and practice coding daily
- Join blockchain communities for support and networking
- Keep your portfolio updated with your projects

Good luck on your journey to becoming a Blockchain Developer! 🚀

► How did I get this answer?

RESULT

Now, I know some basic definitions in blockchain, and I need only to improve my skills in smart contract, give accordingly

Sketch 3 (8:52 PM)

Sure! Here's a personalized roadmap to help you improve your skills in smart contracts. This plan assumes you can dedicate 1-2 hours per day.

Week 1: Advanced Smart Contract Concepts

Topics to Learn:

- Advanced Solidity Features
- Security Best Practices
- Gas Optimization Techniques
- Smart Contract Design Patterns

Tools/Platforms:

- Solidity Documentation
- OpenZeppelin Contracts Library
- Ethernaut: Smart Contract Security Game

Mini-Project:

- Implement a secure and gas-optimized ERC-20 token

Learning Outcome:

Learning Outcome:

- Understand advanced Solidity features
- Implement security best practices
- Optimize gas usage in smart contracts

Optional Certifications:

- Certified Solidity Developer (CSD) by Blockchain Council

Week 2: Smart Contract Testing and Deployment

Topics to Learn:

- Unit Testing with Truffle and Hardhat
- Integration Testing
- Deploying Smart Contracts to Mainnet
- Monitoring and Upgrading Smart Contracts

Tools/Platforms:

- Truffle Suite
- Hardhat
- Etherscan

Mini-Project:

- Write and deploy a smart contract with comprehensive tests

Learning Outcome:

- Write and run tests for smart contracts
- Deploy smart contracts to the Ethereum mainnet

Learning Outcome:

- Write and run tests for smart contracts
- Deploy smart contracts to the Ethereum mainnet
- Monitor and upgrade smart contracts

Optional Certifications:

- Certified Blockchain Developer (CBD) by Blockchain Council

Week 3: Building Decentralized Applications (dApps)

Topics to Learn:

- Frontend Integration with Web3.js
- Backend Integration with Smart Contracts
- User Authentication and Authorization
- dApp Security Best Practices

Tools/Platforms:

- Web3.js
- React.js
- MetaMask

Mini-Project:

- Build a simple dApp with user authentication

Learning Outcome:

- Integrate frontend and backend with smart contracts

Learning Outcome:

- Integrate frontend and backend with smart contracts
- Implement user authentication and authorization
- Ensure dApp security

Optional Certifications:

- Certified dApp Developer (CdD) by Blockchain Council

Week 4: Real-World Projects and Job Preparation

Topics to Learn:

- Blockchain Use Cases
- Industry Trends and Job Market
- Resume and Portfolio Building
- Interview Preparation

Tools/Platforms:

- LinkedIn: Networking and Job Search
- GitHub: Showcase Your Projects
- LeetCode: Practice Coding Interviews

Mini-Project:

- Contribute to an open-source blockchain project

Learning Outcome:

- Gain real-world experience
- Prepare for job interviews

Optional Certifications:

- Certified Blockchain Professional (CBP) by Blockchain Council

Additional Resources:

- **Books:** "Mastering Ethereum" by Andreas M. Antonopoulos, "Solidity Programming Essentials" by Ritabh Modi
- **Communities:** Reddit (r/ethereum, r/blockchain), Stack Overflow, Ethereum Dev Forums

Tips for Success:

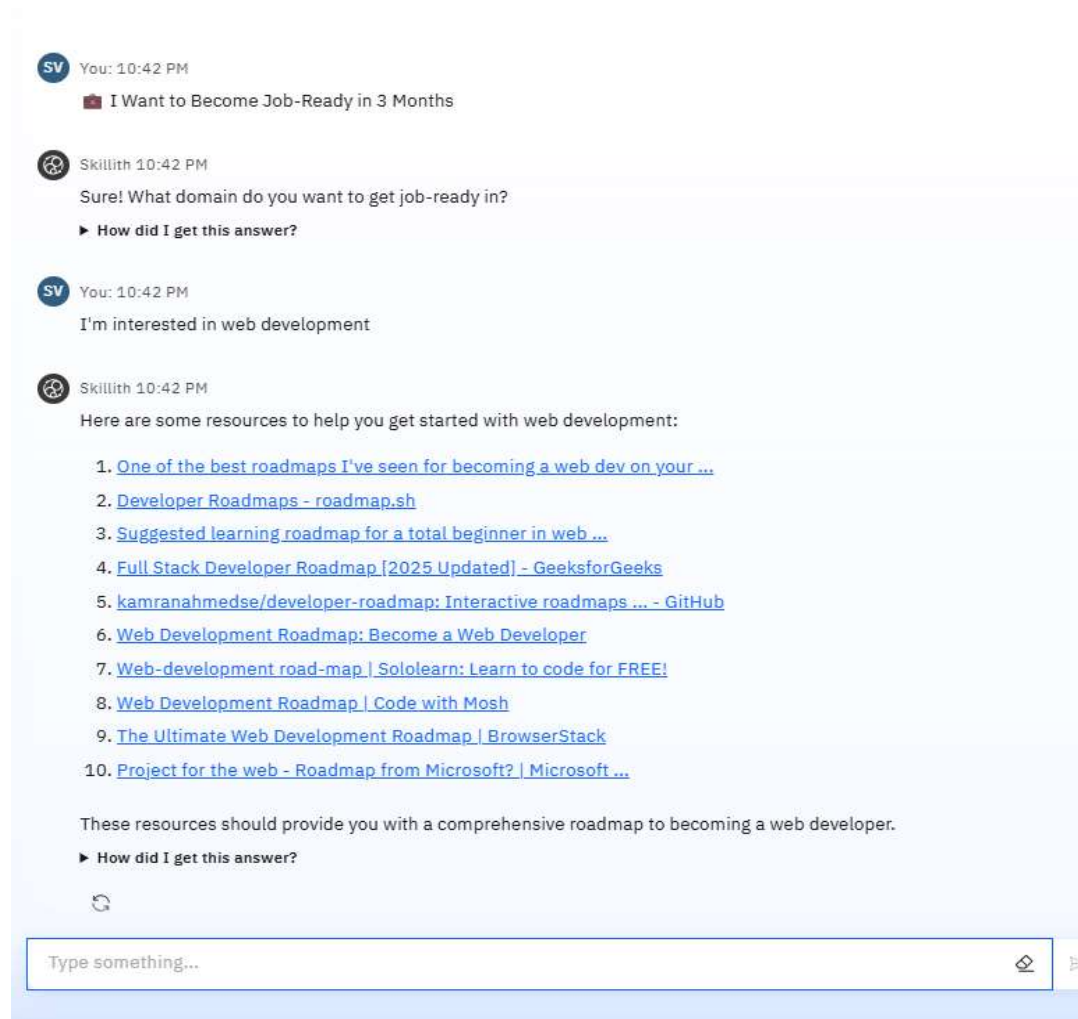
- Stay consistent and practice coding daily
- Join blockchain communities for support and networking
- Keep your portfolio updated with your projects

Good luck on your journey to becoming a Smart Contract Developer! 🚀

► How did I get this answer?



RESULT



RESULT

You: 11:24 PM
React, intermediate, 1 hr daily.

Skash 11:24 PM
Great! Here's a structured learning plan for you:

Week 1: React Basics

Topics:

- Introduction to React
- Setting up a React environment
- JSX basics
- Components and Props

Tools/Platforms:

- React Official Documentation
- CodeSandbox for online coding

Mini-Project:

- Create a simple "To-Do List" app

Learning Outcome:

Learning Outcome:

- Understand the basics of React and how to create simple components

Week 2: State Management

Topics:

- State in React
- Lifting State Up
- Context API

Tools/Platforms:

- React Official Documentation
- CodeSandbox

Mini-Project:

- Enhance the "To-Do List" app with state management

Learning Outcome:

- Understand how to manage state in React applications

Week 3: Advanced React

Topics:

- React Router
- Forms in React
- Authentication

Tools/Platforms:

- React Router Documentation
- CodeSandbox

Mini-Project:

- Create a simple blog with authentication

Learning Outcome:

- Understand how to create multi-page applications and handle forms

Week 4: Testing and Optimization

Topics:

- Unit Testing with Jest and React Testing Library
- Performance Optimization

Tools/Platforms:

- Jest Documentation
- React Testing Library Documentation

Mini-Project:

- Write tests for your blog application

Learning Outcome:

- Understand how to test and optimize React applications

Optional Certifications:

- FreeCodeCamp React Certification
- Coursera React Specialization

Additional Resources:

- React Official Documentation
- FreeCodeCamp React Tutorials

• YouTube Channels: Traversy Media, The Net Ninja

Additional Resources:

- React Official Documentation
- FreeCodeCamp React Tutorials
- YouTube Channels: Traversy Media, The Net Ninja

Tips:

- Practice coding daily
- Join React communities for support
- Build projects to apply what you've learned

Good luck with your learning journey! 🚀

► How did I get this answer?



RESULT



You: 10:47 PM



I'm Here Just for Fun – Tell Me a Joke or Fact



Skillith 10:47 PM

Sure! Here's a fun fact for you: Did you know that the shortest war in history lasted only 38 minutes? It was the Anglo-Zanzibar War, fought between the United Kingdom and Zanzibar on August 27, 1896.

► **How did I get this answer?**



Type something...



CONCLUSION

- The Agentic AI Learning Coach project successfully demonstrates the use of Agentic AI capabilities using IBM Watsonx Studio, Granite models, and RAG knowledge bases. It solves a real problem faced by students: “What should I learn next based on who I am and what I want?”
- The agent was built, tested, and deployed with modular steps, allowing for future updates. The current deployment works well for single-turn and follow-up interactions. The project meets the IBM Cloud Lite and IBM Granite usage requirements. With minimal enhancements like UI integration and progress tracking, this system can easily evolve into a complete SaaS product to help students and learners worldwide.

FUTURE SCOPE

Due to time limitations, the frontend UI and backend hosting were not completed. However, plans are in place for the following:

- A clean web-based chat UI where users can provide their inputs
- Backend (Flask or Node.js) to connect the frontend with the Watsonx API
- MongoDB or Cloudant for storing user inputs and tracking progress
- Option to resume conversations
- Authenticated user dashboard to manage roadmaps
- All of these can be deployed using IBM Cloud services.

REFERENCES

- **IBM Watsonx Documentation**

The official IBM documentation was the core reference for setting up Watsonx Studio, creating AI agents, integrating RAG (Retrieval-Augmented Generation), and deploying models like IBM Granite. It provided all necessary steps and configuration guidelines.

- <https://www.ibm.com/docs/en/watsonx>.

- **Zoom Session Recordings (Internal)**

Live classes and recorded Zoom sessions played a crucial role in understanding practical implementation details, especially around deploying AI agents, configuring runtimes, and testing use cases in real-time using Watsonx Studio.

- **YouTube – IBM Technology Channel**

Video tutorials from IBM's official YouTube channel helped visualize the agent creation process, explore model behavior (Granite, Mistral), and understand how to deploy working agents using the Watsonx interface.

- <https://www.youtube.com/@IBMTechology>

IBM CERTIFICATIONS

In recognition of the commitment to achieve
professional excellence



Sai Videsh

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence



Issued on: Jul 21, 2025
Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/84df3fc9-7e0c-41b4-b71f-82cacc10b7de>



IBM CERTIFICATIONS



IBM CERTIFICATIONS

IBM **SkillsBuild**

Completion Certificate



This certificate is presented to

Sai Videsh

for the completion of

**Lab: Retrieval Augmented Generation with
LangChain**

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 26 Jul 2025 (GMT)

Learning hours: 20 mins



THANK YOU