

***M.A.M COLLEGE OF ENGINEERING AND TECHNOLOGY***

Siruganur, Tiruchirappalli.

Department of

Artificial IntelligenceAnd Data Science

A Project on

***Optimizing User, Group, and Role Management with Access Control and Workflows***

***NM ID:***

***TEAM ID :*** NM2025TMID04962

***TEAM SIZE :*** 05

***GROUP MEMBER’S NAME WITH REG. NO :***

MOHAMMED GANI .H - 812022243031(TEAM LEADER)

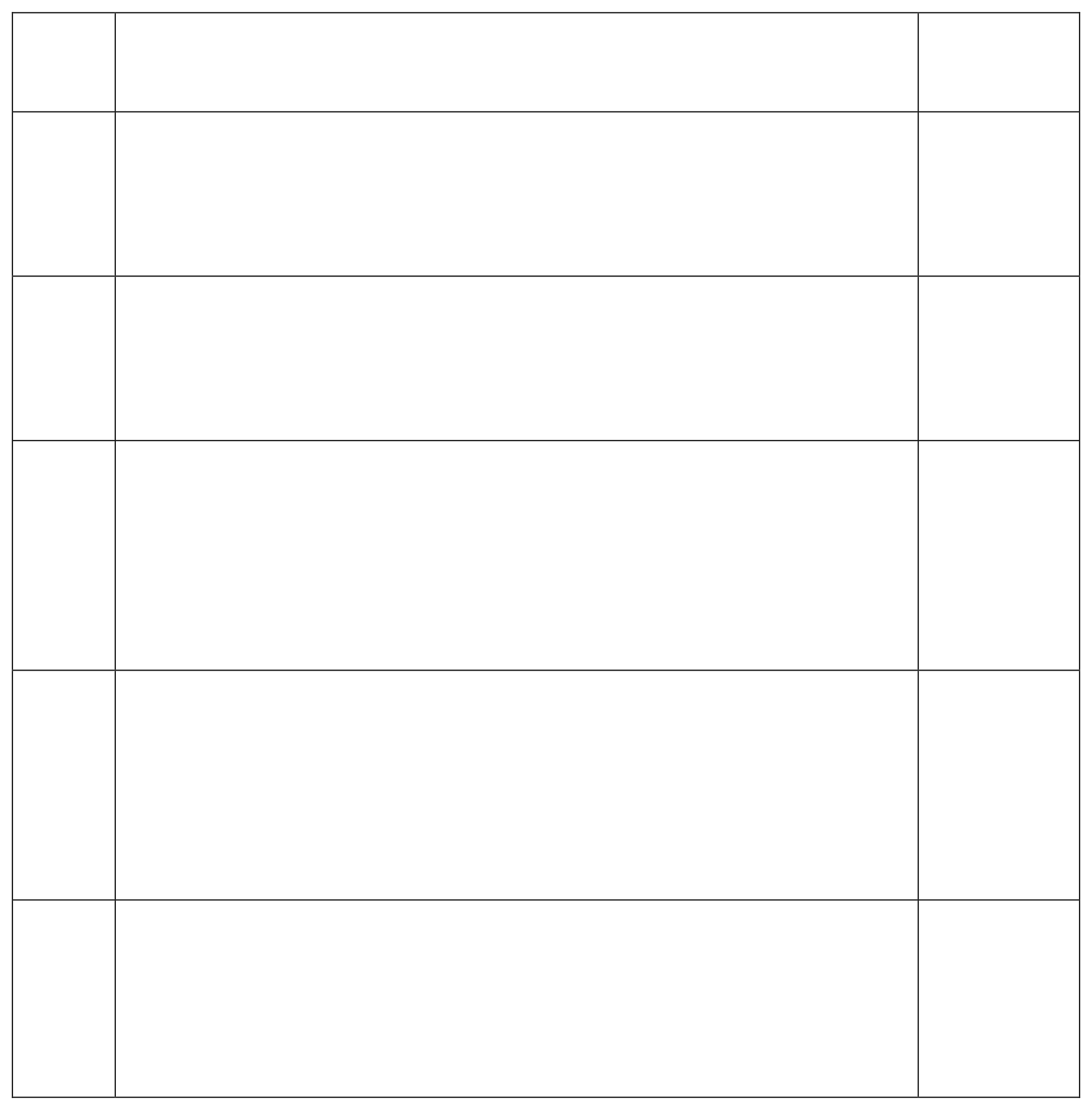
RAGUL .N- 812022243040 (TEAM MEMBER)

PRABHAKARAN .K -812022243036(TEAM MEMBER)

SANJAI .B -812022243047 (TEAM MEMBER)

MUHIL RAJ .N-812022243701 (TEAM MEMBER)

### TABLE OF CONTENTS



S.NO

CONTENT

PAGE NO

1

IDEATION PHASE

Problem Statement Empathy Map Canvas

4

6

2

PERFORMANCE & TESTING

Procedure/ Implementation Steps Testing Screenshots

9

10

14

3

PROJECT DESGIN PHASE

Problem Solution fit Proposed Solution Conclusion

Solution Architecture

33

34

35

36

37

4

PROJECT PLANNING PHASE

Product Backlog Sprint Planning User Stories

38

39

40

Story Points

5

REQUIREMENT ANALYSIS

Solution Requirement Dataflow Diagram Technology Stack

41

42

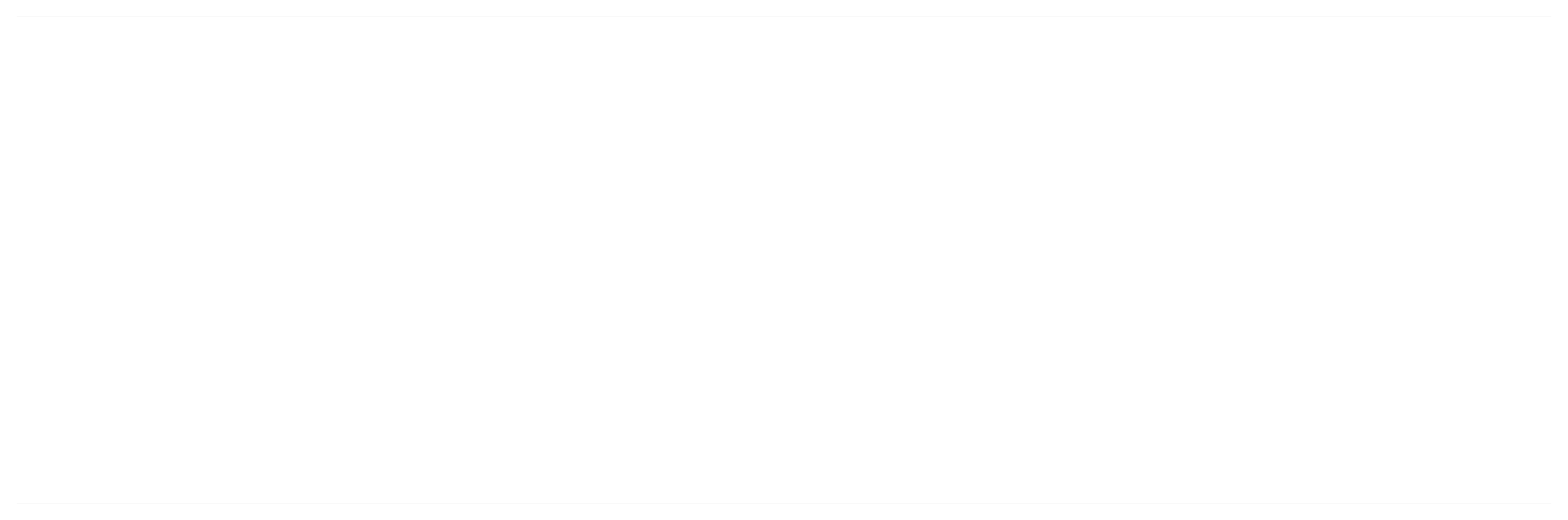
43

44

2

# IDEATION PHASE

3



**Problem Statement**

In a small project management team consisting of a Project Manager (Alice) and a Team Member (Bob), there is a need to efficiently manage project tasks and ensure accountability throughout the project lifecycle. The current system lacks clear role definitions, access controls, and a structured workflow, leading to confusion regarding task assignments and progress tracking.

**Problem Definition**

In small project teams, effective coordination and role clarity are essential for smooth task execution and timely project completion. However, the current working process between the Project Manager (Alice) and the Team Member (Bob) lacks a structured system for managing tasks, tracking progress, and maintaining accountability. The absence of defined roles, access control, and workflow structure often leads to confusion, duplication of work, and communication gaps.

Without a centralized platform, team members struggle to identify responsibilities, monitor task progress, and ensure deadlines are met. Both users currently have unrestricted access to modify or update tasks, leading to inconsistencies and inefficiencies in project execution.

Therefore, there is a need for a role-based project management system that defines clear responsibilities and access levels for each team member. The system should streamline task assignment, progress monitoring, and reporting, ensuring transparency and accountability throughout the project lifecycle. By addressing these issues, the proposed system aims to enhance productivity, reduce miscommunication, and improve overall project performance.

### Abstract

Effective project management is essential for ensuring smooth task execution and accountability within teams. In a small project management team comprising a Project Manager (Alice) and a Team Member (Bob), the absence of a structured system has led to confusion in task assignments, unclear roles, and poor progress tracking. To overcome these challenges, this project proposes a role-based project management system that defines user roles, enforces access control, and streamlines task workflows.

The Project Manager can assign and monitor tasks, while the Team Member can view and update only their assigned work, ensuring accountability and transparency. The system also includes features such as task tracking, progress monitoring, and notifications for efficient collaboration. By introducing structured workflows and clear responsibilities, the system

enhances productivity, minimizes miscommunication, and promotes efficient project execution for small teams.

4

**Empathy Map canvas**

## User 1: Project Manager (Alice)

Section

Thinks

Says

Does

Feels

Description

“I need a clear overview of all project tasks.” “I want to ensure that everyone is accountable and deadlines are met.” “It’s difficult to track progress without a proper system.”

“Who’s responsible for this task?” “We’re missing deadlines because updates are not properly tracked.” “I need status reports in one place.”

Assigns tasks manually, follows up via calls or chats, updates project notes in separate tools. Spends extra time monitoring progress manually.

Frustrated due to lack of visibility and control. Overwhelmed managing multiple updates. Wants a structured, automated solution for better efficiency.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**User 2: Team Member (Bob)**

Section

Thinks

Says

Does

Feels

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

5

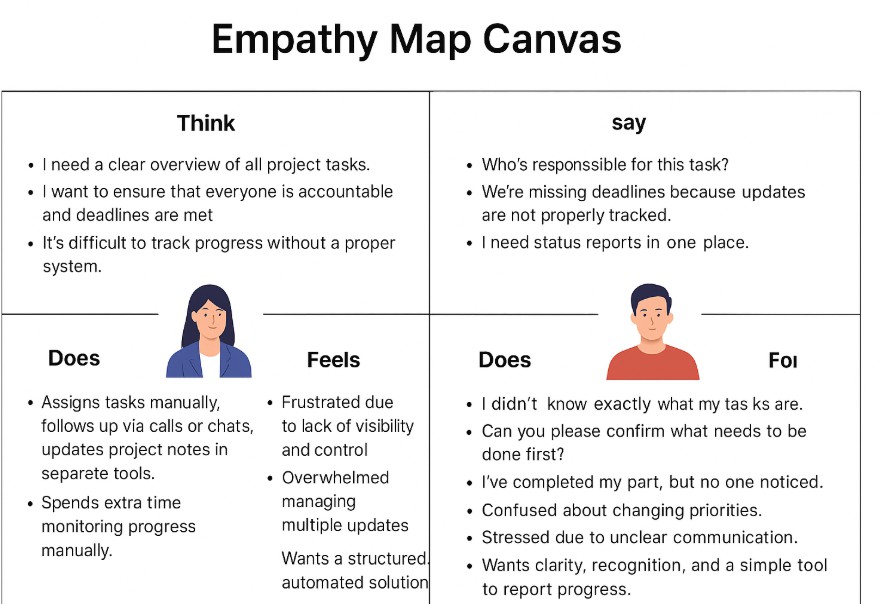
Description

“I need to know exactly what my tasks are.” “I wish there was one place to update my work and check deadlines.” “Sometimes tasks get mixed up due to unclear instructions.”

“I didn’t know the task priority.” “Can you please confirm what needs to be done first?” “I’ve completed my part, but no one noticed.”

Completes assigned work, gives verbal updates, forgets to record progress formally, depends on manager’s follow-up.

Confused about changing priorities. Stressed due to unclear communication. Wants clarity, recognition, and a simple tool to report progress.



6

### Introduction

Project management plays a vital role in ensuring that tasks are completed efficiently, on time, and within scope. In small teams, effective coordination and communication are essential to maintain productivity and achieve project goals. However, when there is no structured system in place, managing tasks, defining responsibilities, and tracking progress can become challenging. This project focuses on addressing these challenges faced by a small team consisting of a Project Manager (Alice) and a Team Member (Bob).

Currently, the team lacks a defined workflow and proper access control mechanisms, leading to confusion over task ownership and delays in project completion. Both members often face difficulties in identifying who is responsible for specific tasks, resulting in miscommunication and lack of accountability. To resolve this, the proposed system introduces a role-based project management solution that clearly defines user roles and permissions.

In this system, the Project Manager has administrative rights to create, assign, and monitor tasks, while the Team Member has limited access to update and report progress on assigned tasks. Additional features such as task tracking, progress visualization, and automated notifications will enhance coordination and transparency within the team.

By implementing this structured approach, the project aims to streamline communication, improve accountability, and enhance productivity in small project environments. Ultimately, this solution provides a simple yet effective framework for managing projects efficiently, ensuring clarity, and fostering collaboration between team members.

### Objectives

The main objective of this project is to design and implement a role-based project management system that enhances efficiency, transparency, and accountability within small teams. The system focuses on addressing the lack of structured workflows, undefined roles, and inefficient task management in the current working process.

1. To establish clear user roles and permissions
2. Define distinct roles for the Project Manager and Team Member to ensure proper task ownership and responsibility. The Project Manager will have full control over creating, assigning, and monitoring tasks, while the Team Member will have limited access to view and update assigned tasks.
3. To streamline task assignment and workflow management
4. Develop a structured mechanism for task creation, prioritization, and tracking to prevent overlapping responsibilities and confusion. This ensures that every task is properly documented, assigned, and monitored.
5. To provide real-time task and progress tracking
6. Incorporate a dashboard that displays the real-time status of ongoing tasks, completion percentages, and upcoming deadlines, enabling both team members to stay informed and aligned with project goals.
7. To enhance communication and collaboration
8. Facilitate smooth interaction between team members through automated notifications, reminders, and updates whenever a task is created, modified, or completed.

9 .To improve accountability and transparency

1. Maintain an activity log to record all task-related actions, ensuring that every update is traceable and measurable. This helps in evaluating individual performance and maintaining transparency.
2. To generate analytical reports and insights
3. Implement reporting features that summarize task progress, completion rate, and workload distribution to help the Project Manager make data-driven decisions.
4. To increase overall project efficiency and productivity
5. By integrating all the above features, the system aims to reduce manual effort, eliminate communication gaps, and ensure that projects are completed on time with improved coordination.

### Scope

The scope of this project focuses on developing a role-based project management system designed to improve coordination, accountability, and task management within small project teams. The system primarily addresses the needs of a two-member team comprising a Project Manager (Alice) and a Team Member (Bob), but it can be easily scaled to support larger teams in the future.

This project covers the design, development, and implementation of a digital platform where tasks can be created, assigned, tracked, and updated based on user roles. The Project Manager will have administrative privileges to manage tasks, monitor progress, and generate reports, while the Team Member will have restricted access limited to viewing and updating their assigned tasks.

The system will include core functionalities such as task creation, progress tracking, notifications, reporting, and activity logging. It will provide a user-friendly interface that ensures smooth navigation and efficient workflow management. The solution will also support real-time updates to help team members stay informed about deadlines and changes.

Out of Scope

Enterprise-Level Integration:

The system is not designed to handle large-scale enterprise environments involving multiple departments, external collaborators, or complex project hierarchies.

Advanced AI and Cloud Automation:

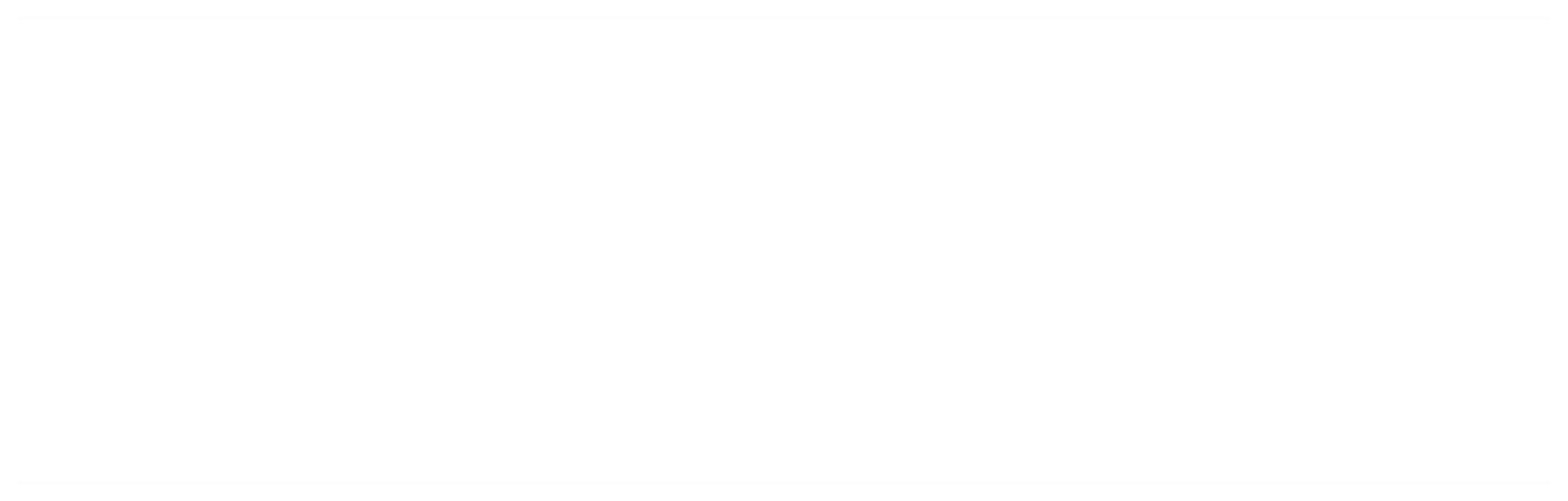
Features such as artificial intelligence-based task prediction, cloud synchronization, and automated workflow optimization are not included in the current phase and are planned for future enhancement.

# PERFORMANCE & TESTING

9



***Procedure or Implementation steps***

***Phase 1 : Create Users***

1. Open service now
2. Click on All >> search for users
3. Select Users under system security
4. Click on new
5. Fill the following details to create a new user
6. Click on submit
7. Create one more user:
8. Create another user with the following details
9. Click on submit



***Phase 2 : Create Groups***

1. Open service now.
2. Click on All >> search for groups
3. Select groups under system security
4. Click on new
5. Fill the following details to create a new group
6. Click on submit

***Phase 3 : Create Roles***

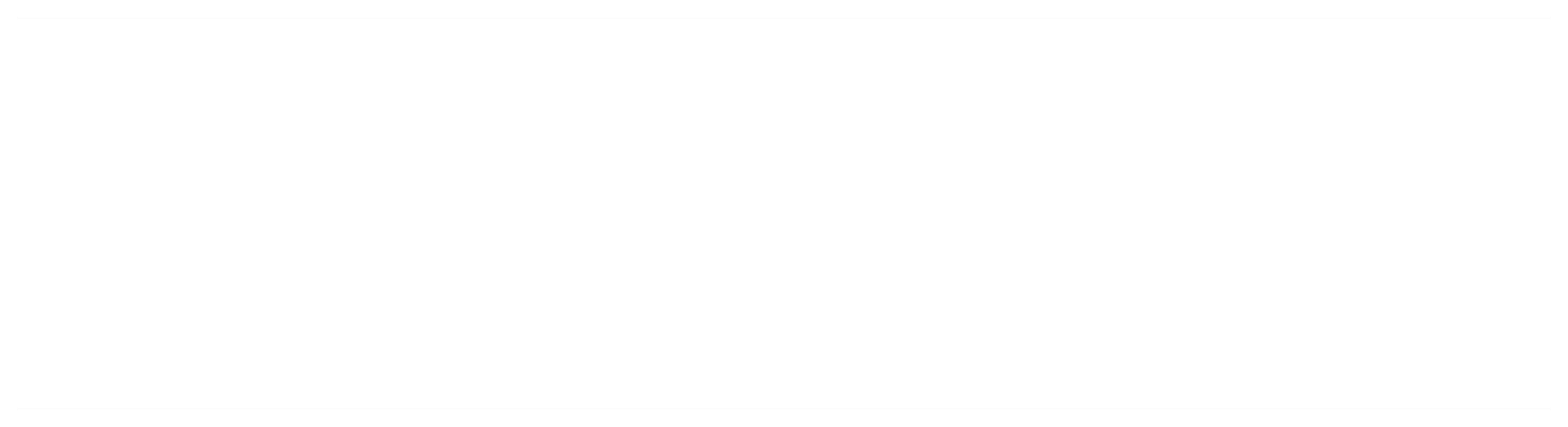
1. Open service now.
2. Click on All >> search for roles
3. Select roles under system security
4. Click on new
5. Fill the following details to create a new role
6. Click on submit

### Create one more role:

1. Create another role with the following details
2. Click on submit

***Phase 4 : Assign roles to alice user***

1. Open servicenow.Click on All >> search for user
2. Select tables under system definition
3. Select the project manager user
4. Under project manager
5. Click on edit
6. Select project member and save
7. click on edit add u\_project\_table role and u\_task\_table role
8. click on save and update the form.



1. Open servicenow.Click on All >> search for user
2. Select tables under system definition
3. Select the bob p user
4. Under team member
5. Click on edit
6. Select team member and give table role and save
7. Click on profile icon Impersonate user to bob
8. We can see the task table2.

***Phase 5 : Assign roles to bob user***

***Phase 6 : Assign table access to application***

1. While creating a table it automatically create a application and module for that table
2. Go to application navigator search for search project table application
3. Click on edit module
4. Give project member roles to that application
5. Search for task table2 and click on edit application.
6. Give the project member and team member role for task table 2 application

***Phase 7:Create ACL***

1. Open service now.
2. Click on All >> search for ACL
3. Select Access Control(ACL) under system security
4. Click on elevate role
5. Click on new
6. Fill the following details to create a new ACL
7. Scroll down under requires role
8. Double click on insert a new row
9. Give task table and team member role
10. Click on submit
11. Similarly create 4 acl for the following fields
12. Click on profile on top right side
13. Click on impersonate user
14. Select bob user
15. Go to all and select task table2 in the application menu bar
16. Comment and status fields are have the edit access

***Phase 8 : Create a Flow to Assign operations ticket to group***

* 1. Open service now.
  2. Click on All >> search for Flow Designer
  3. Click on Flow Designer under Process Automation.
  4. After opening Flow Designer Click on new and select Flow.
  5. Under Flow properties Give Flow Name as “ task table”. 6. Application should be Global.

7. Click build flow.

## Next step:

1. Click on Add a trigger
2. Select the trigger in that Search for “create record” and select that.
3. Give the table name as “ task table ”.
4. Give the Condition as Field : status Operator :is Value : in progress
5. Field : comments Operator :is Value : feedback
6. Field : assigned to Operator :is Value : bob
7. After that click on Done.

## Next step:

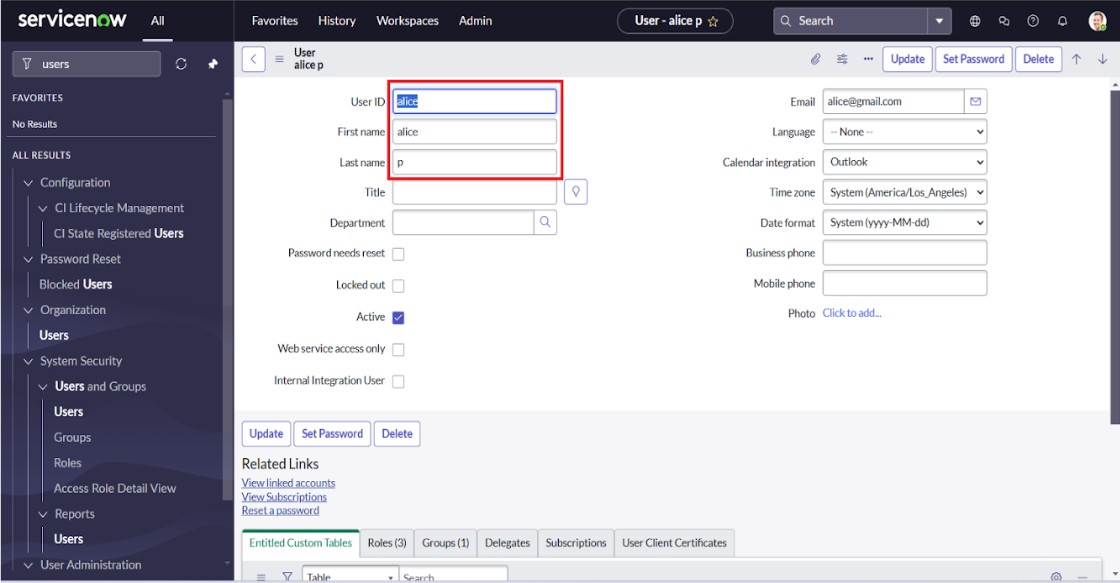
1. Click on Add an action.
2. Select action in that ,search for “ update records”.
3. In Record field drag the fields from the data navigation from Right Side(Data pill)
4. Table will be auto assigned after that
5. Add fields as “status” and value as “completed”
6. Click on Done.

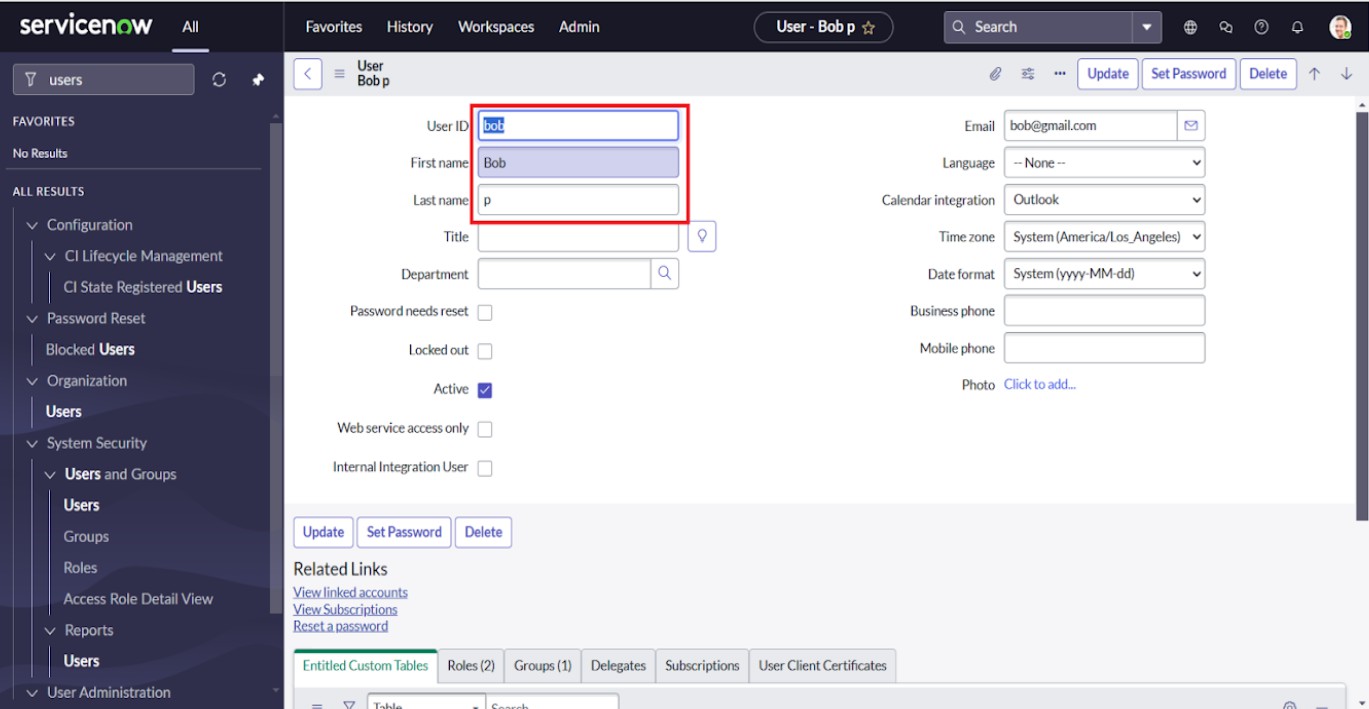
## Next step:

1. Now under Actions.
2. Click on Add an action.
3. Select action in that ,search for “ ask for approval ”.
4. In Record field drag the fields from the data navigation from Right side
5. Table will be auto assigned after that
6. Give the approve field as “ status”
7. Give approver as alice p
8. Click on Done.
9. Go to application navigator search for task table.
10. It status field is updated to completed
11. Go to application navigator and search for my approval
12. Click on my approval under the service desk.
13. Alice p got approval request then right click on requested then select approved

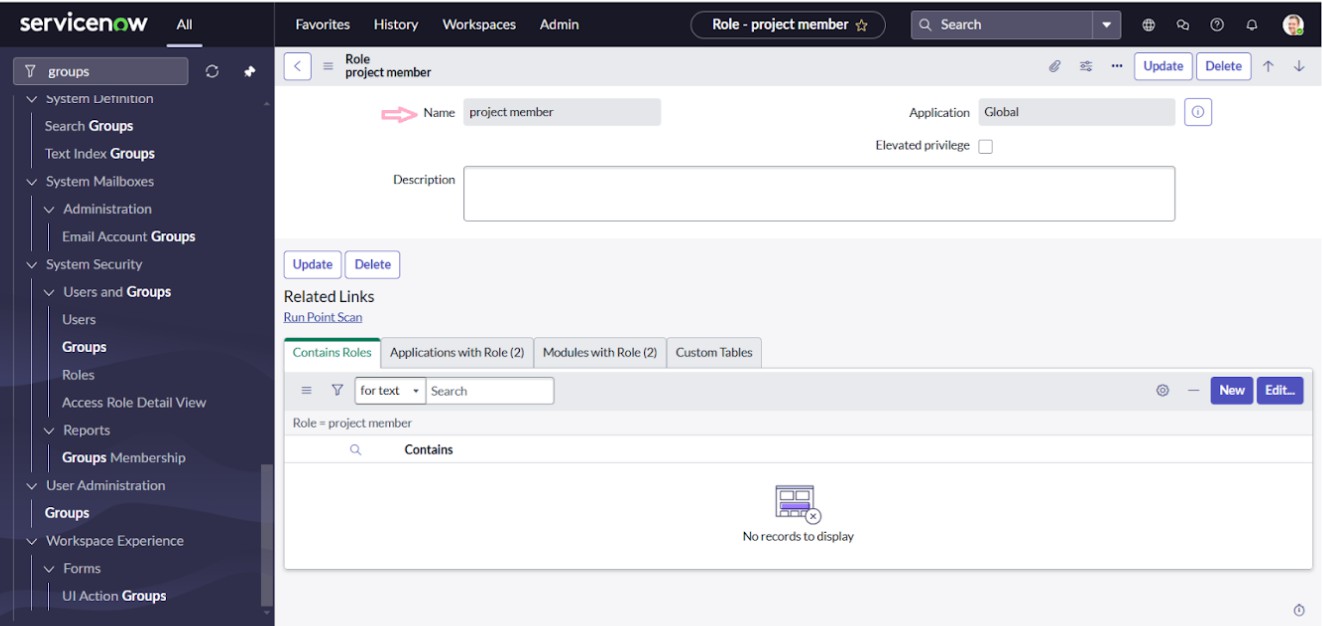
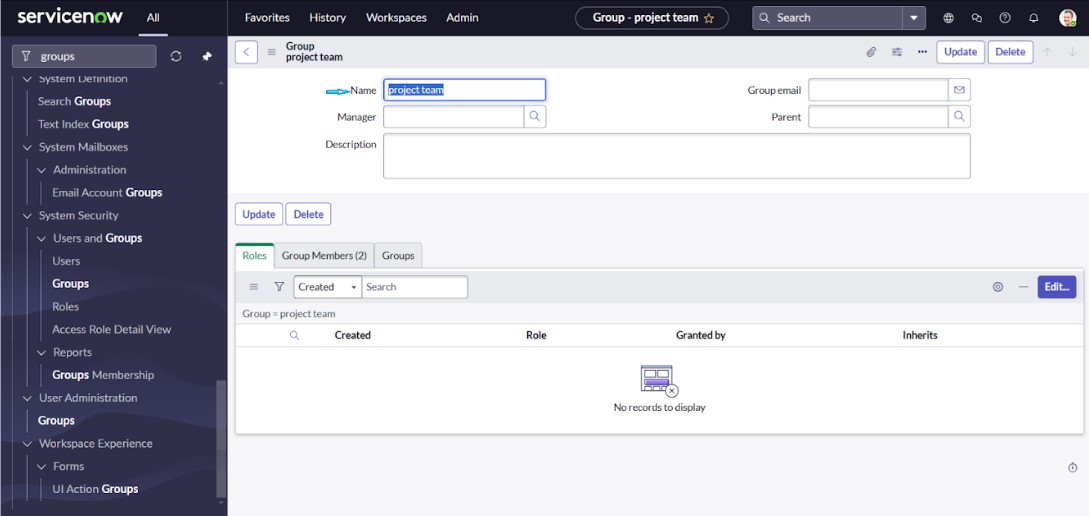
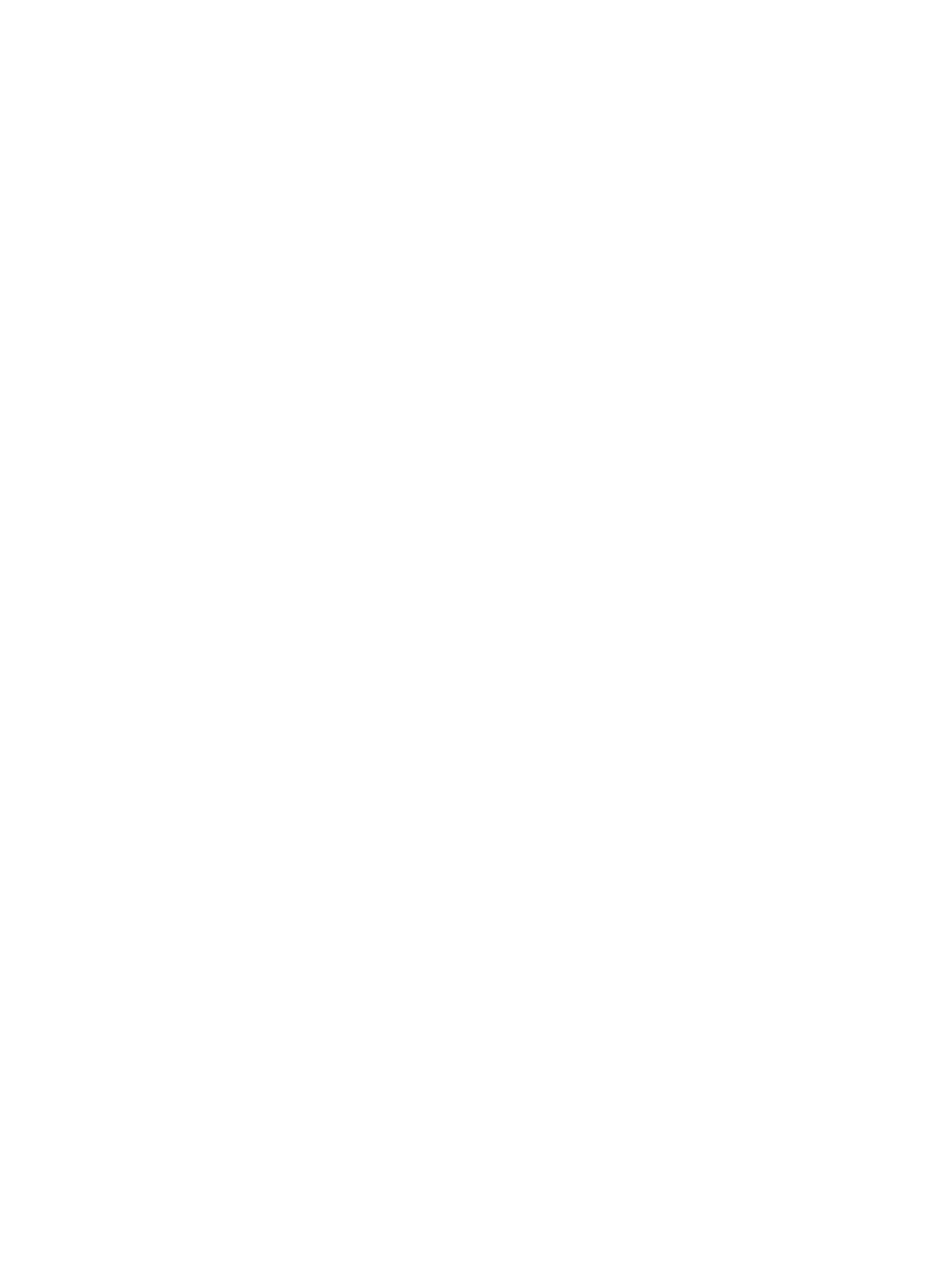
***Screenshots:***

***Phase 1 : Create Users***

******

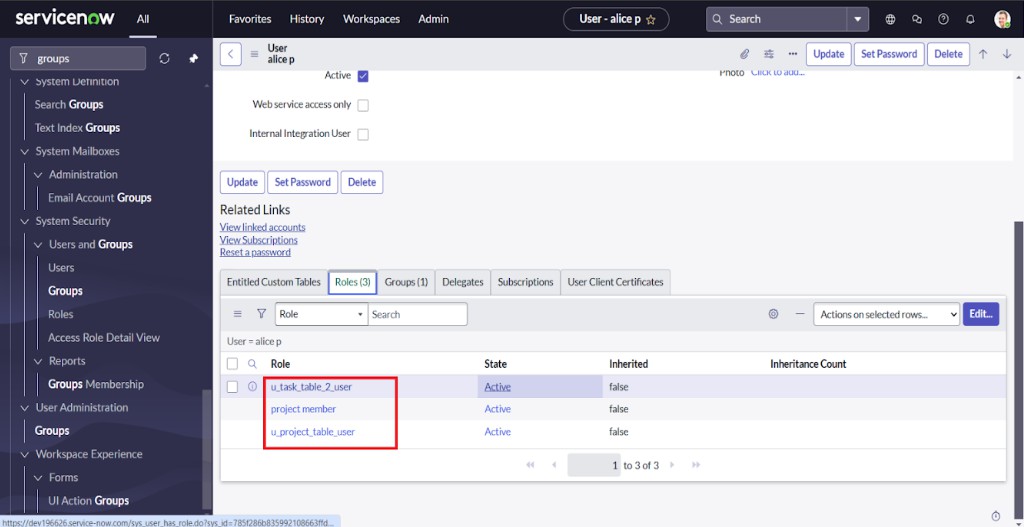
******

***Phase 2 : Create Groups***

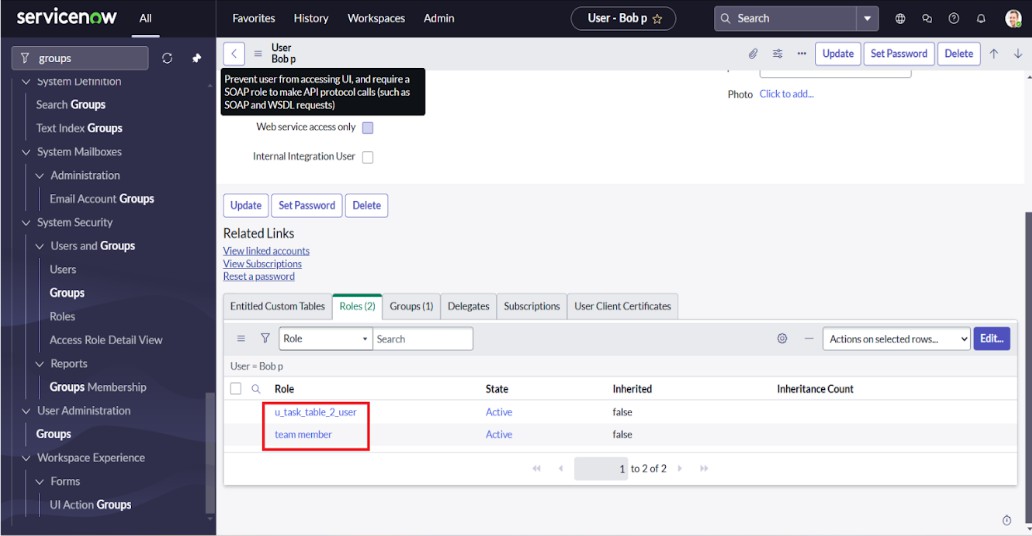


***Phase 3 : Create Roles***

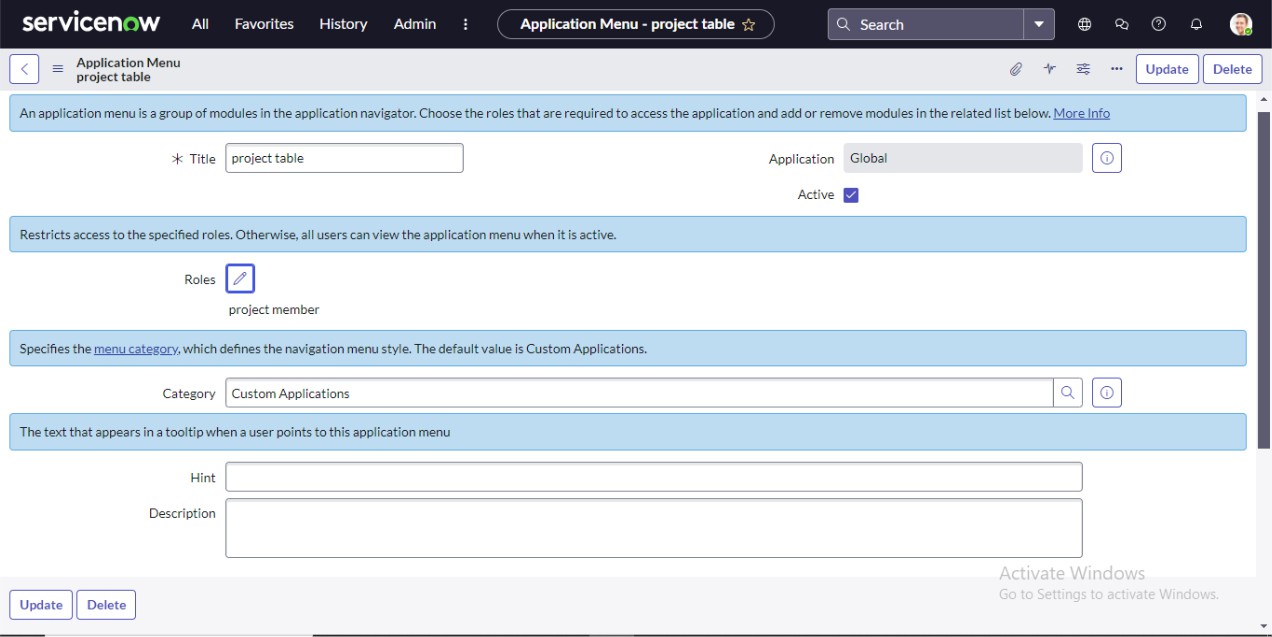
***Phase 4 : Assign roles to alice user***

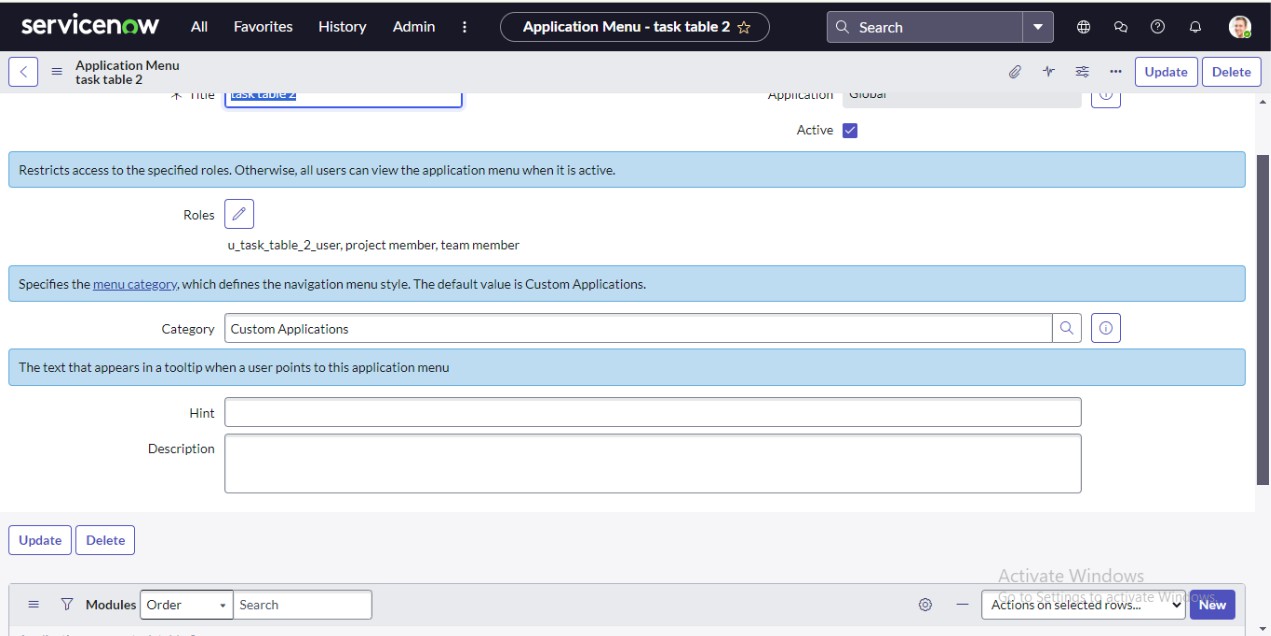
******

***Phase 5 : Assign roles to bob user***

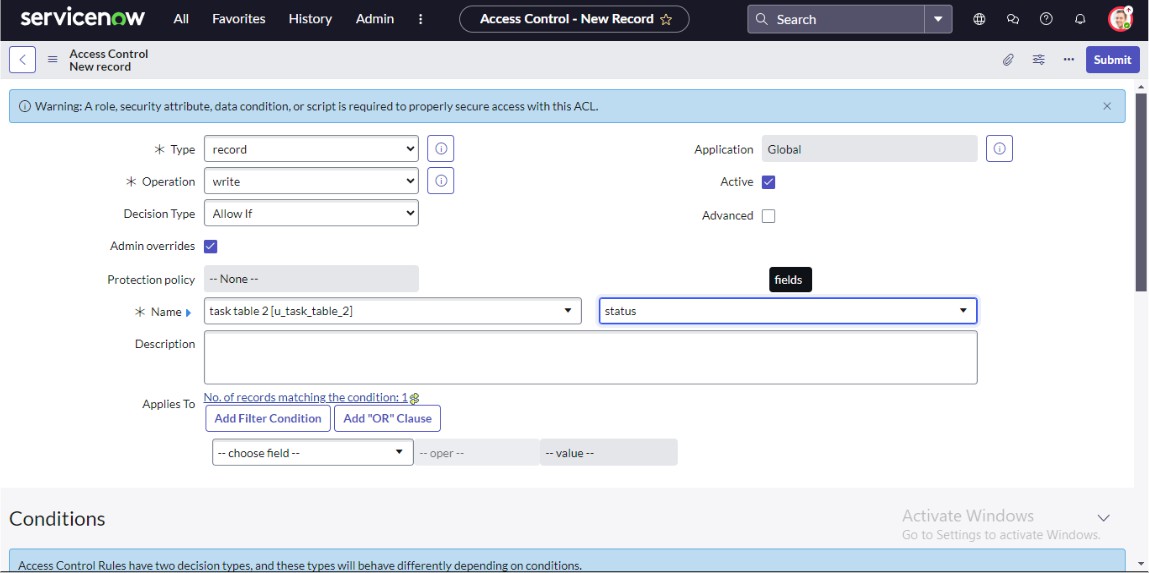
******

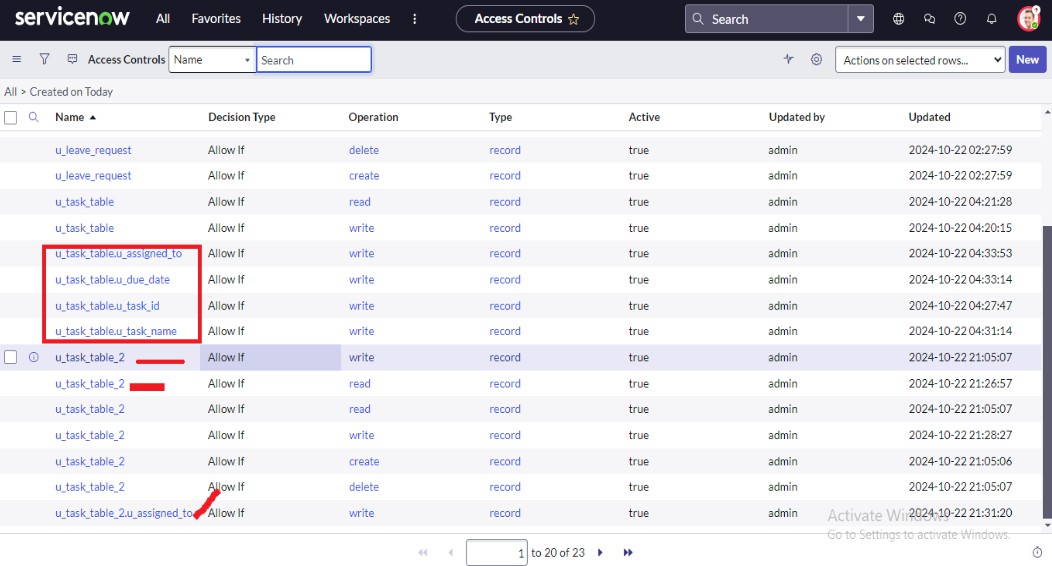
***Phase 6 :Assign table access to application***

******

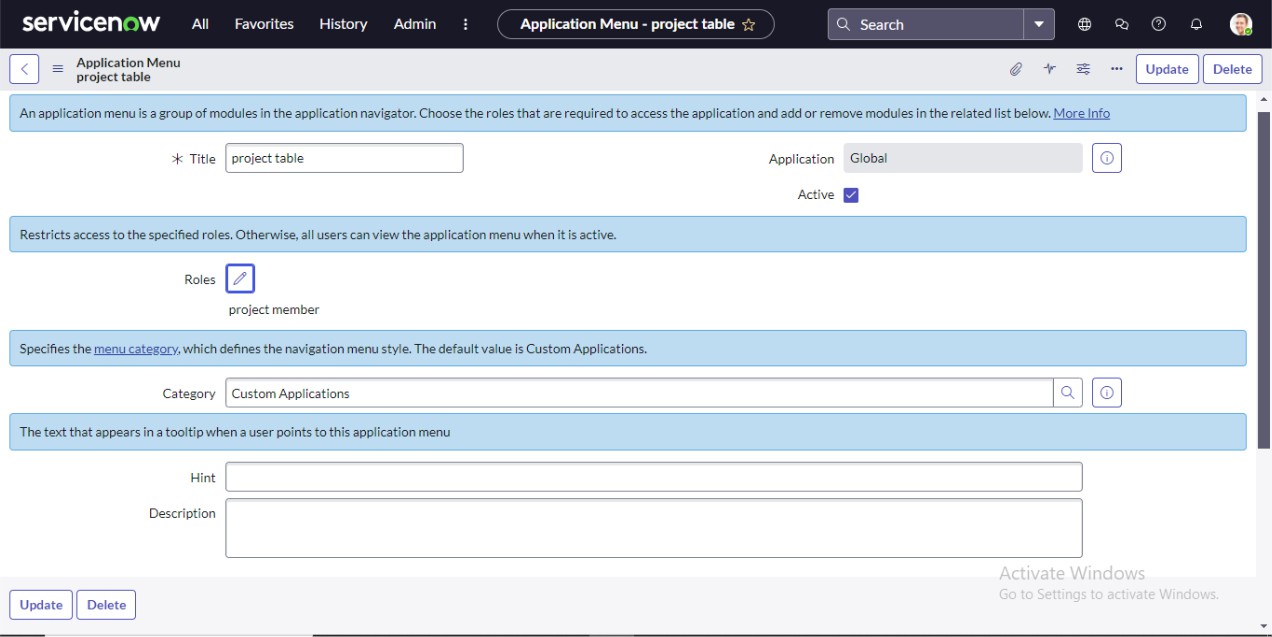
******

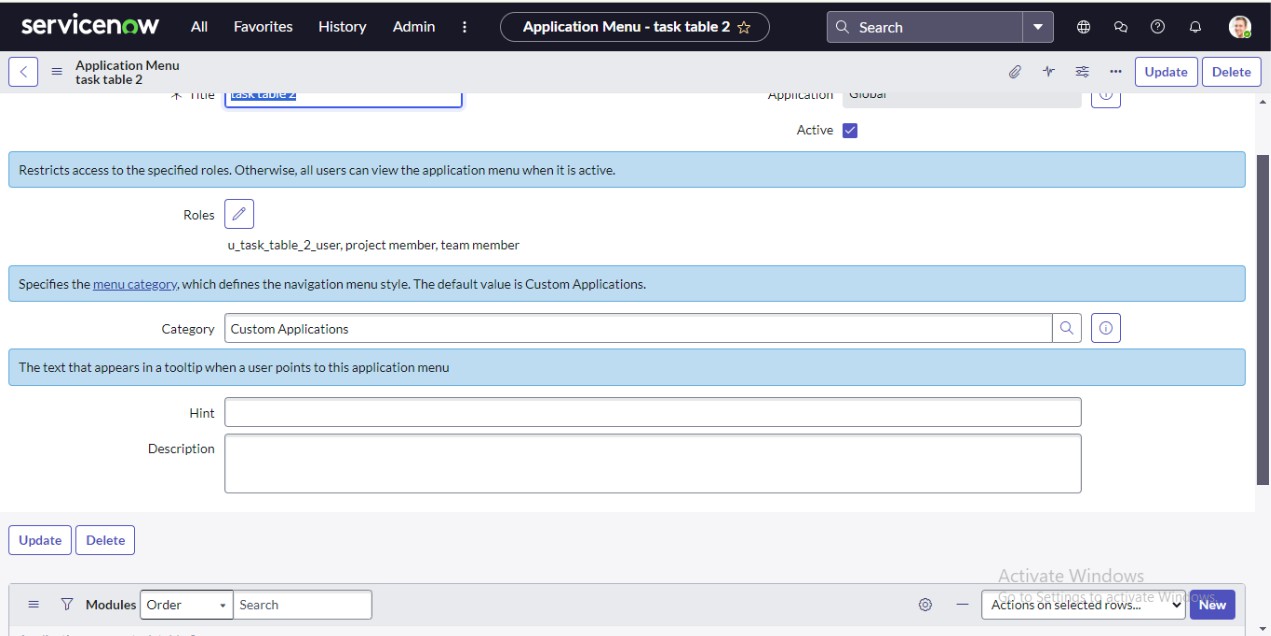
***Phase 7 :Create ACL***

******

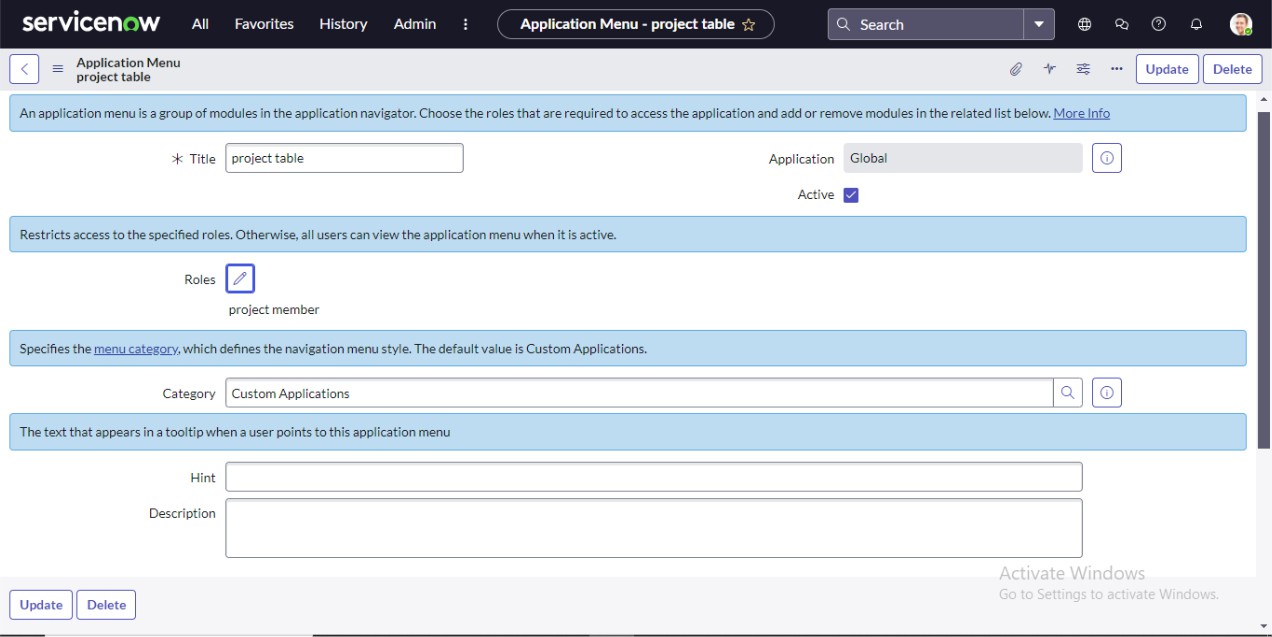


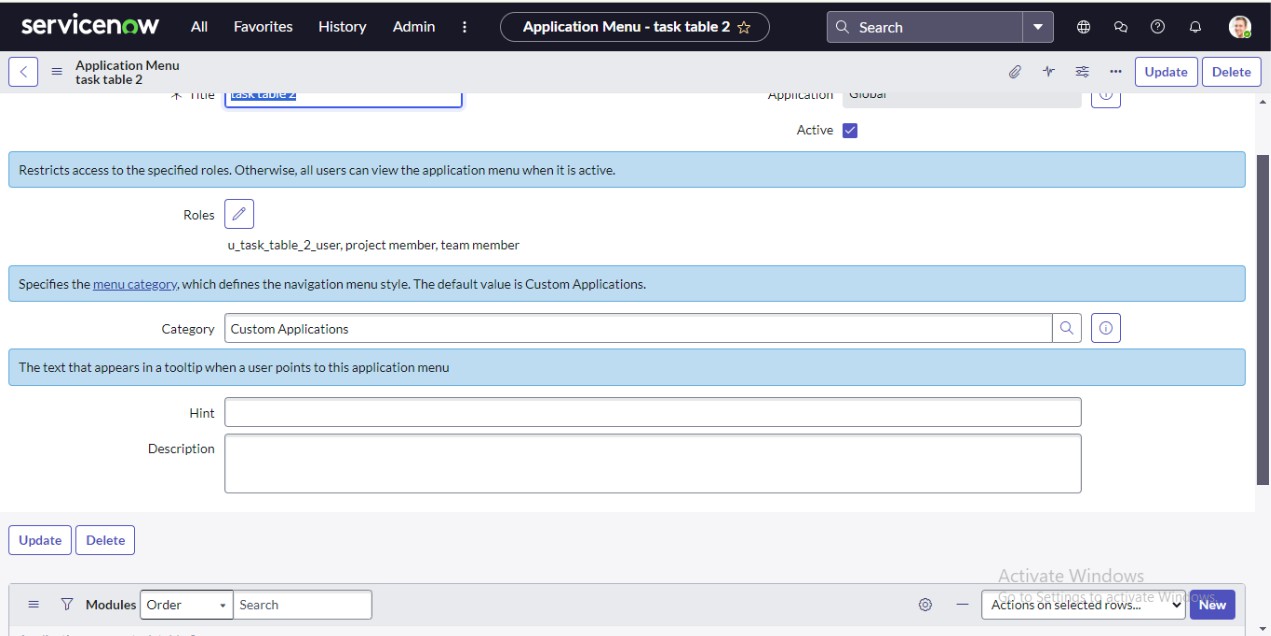
***Phase 6 :Assign table access to application***

******

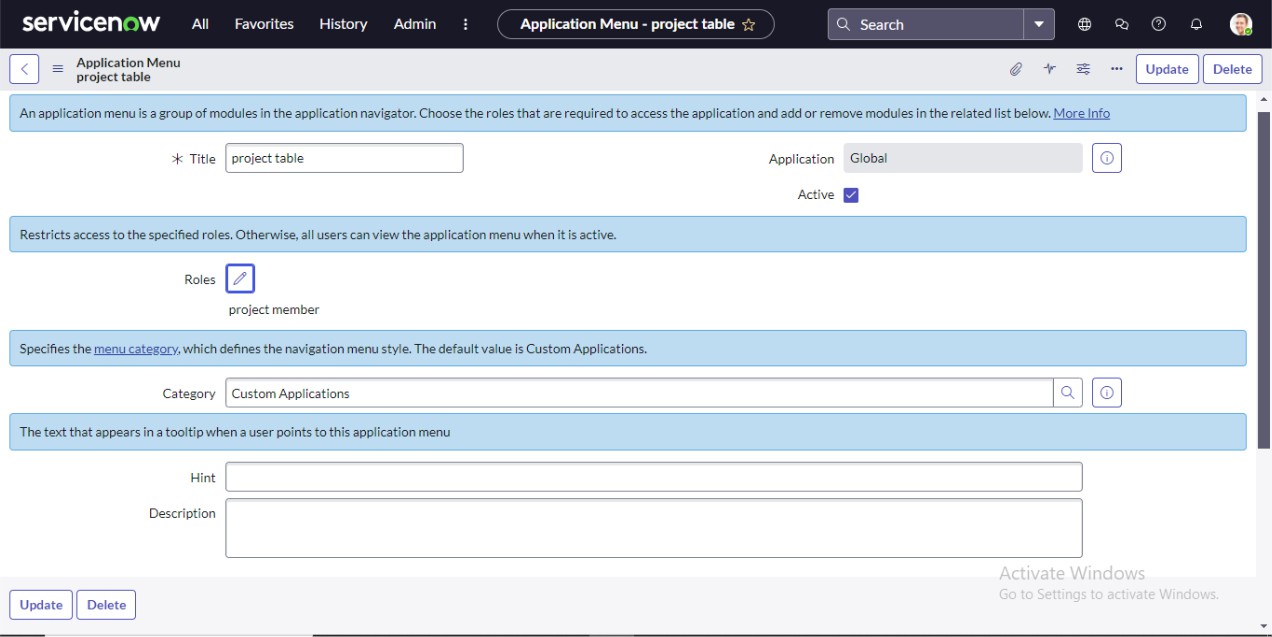
******

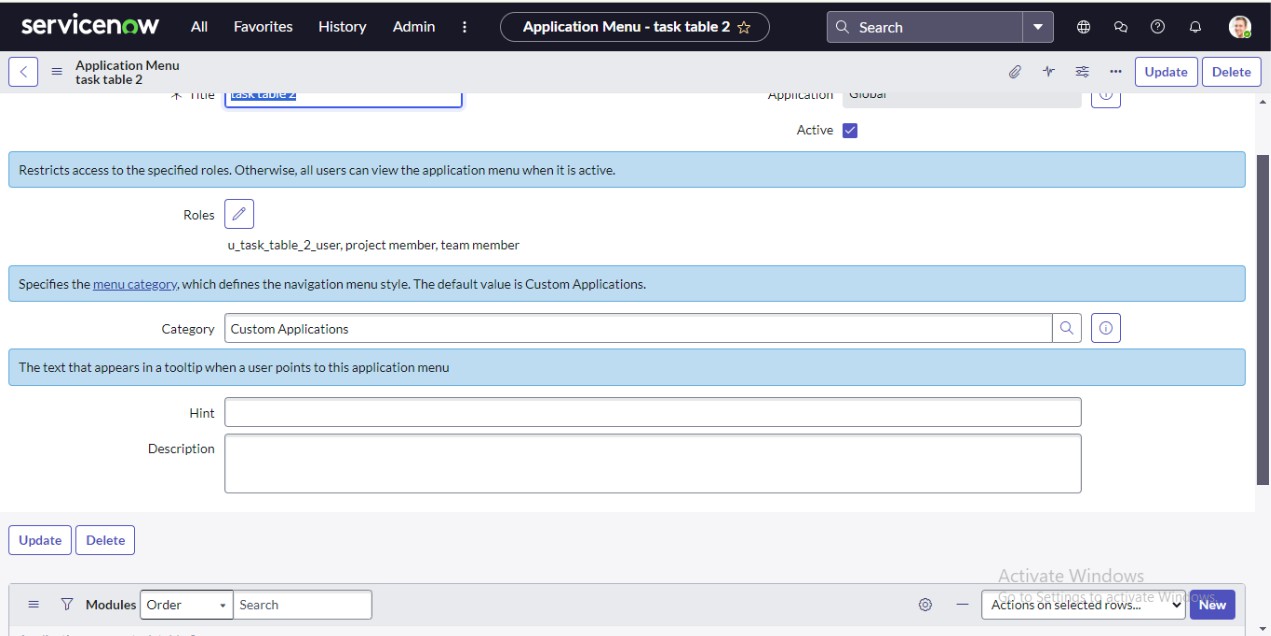
***Phase 6 :Assign table access to application***

******

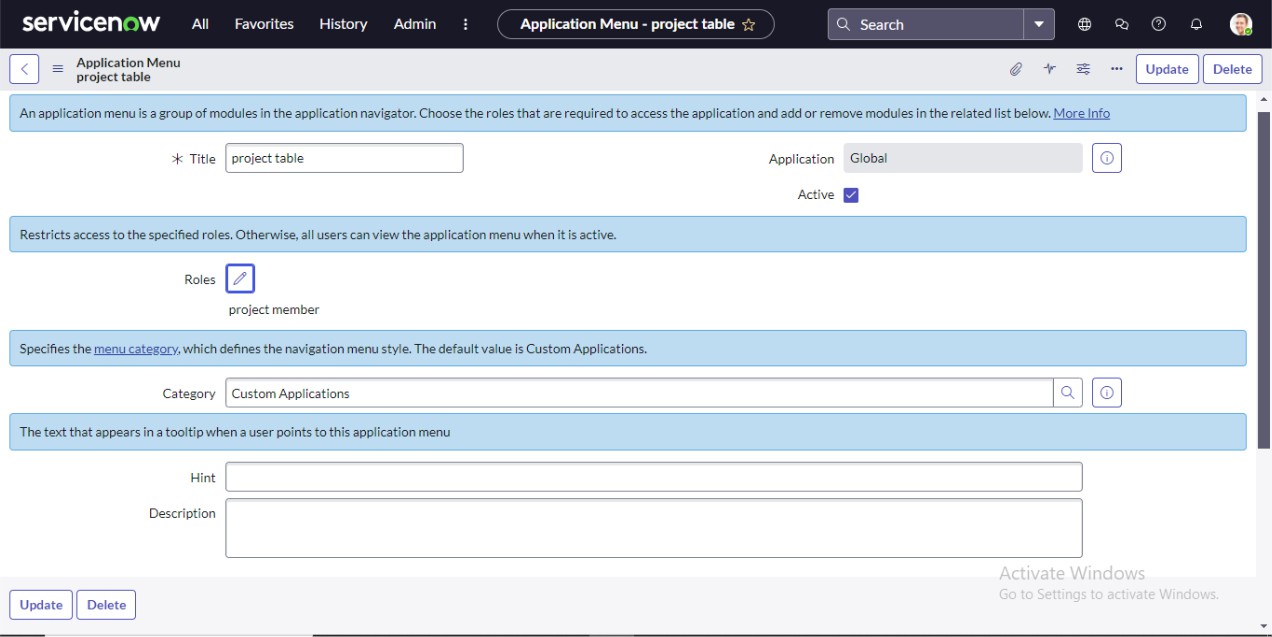
******

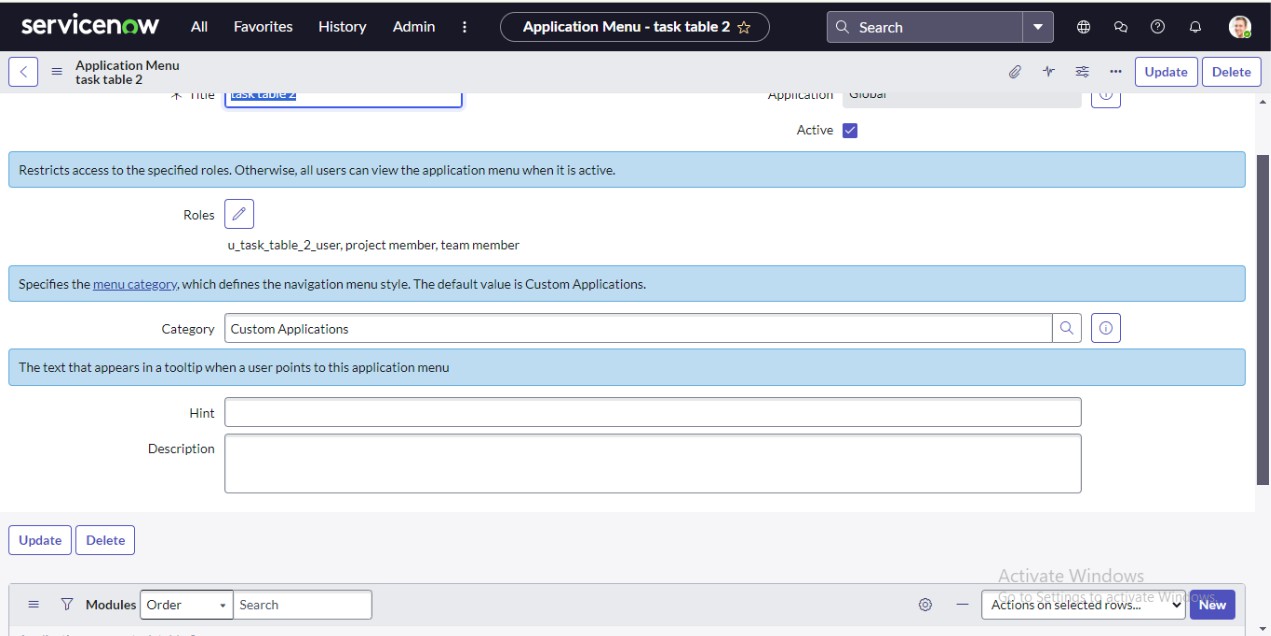
***Phase 6 :Assign table access to application***

******

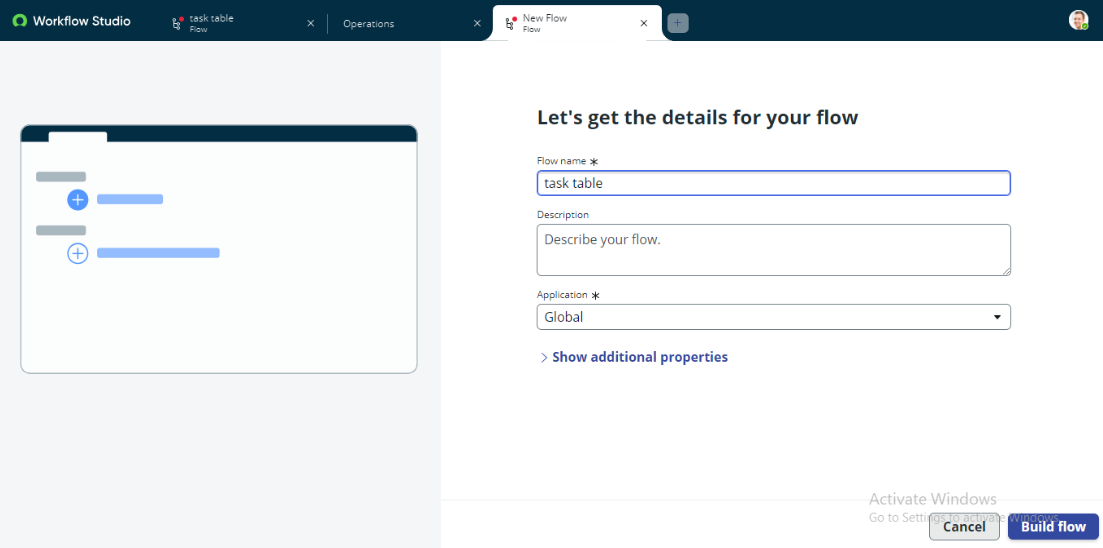
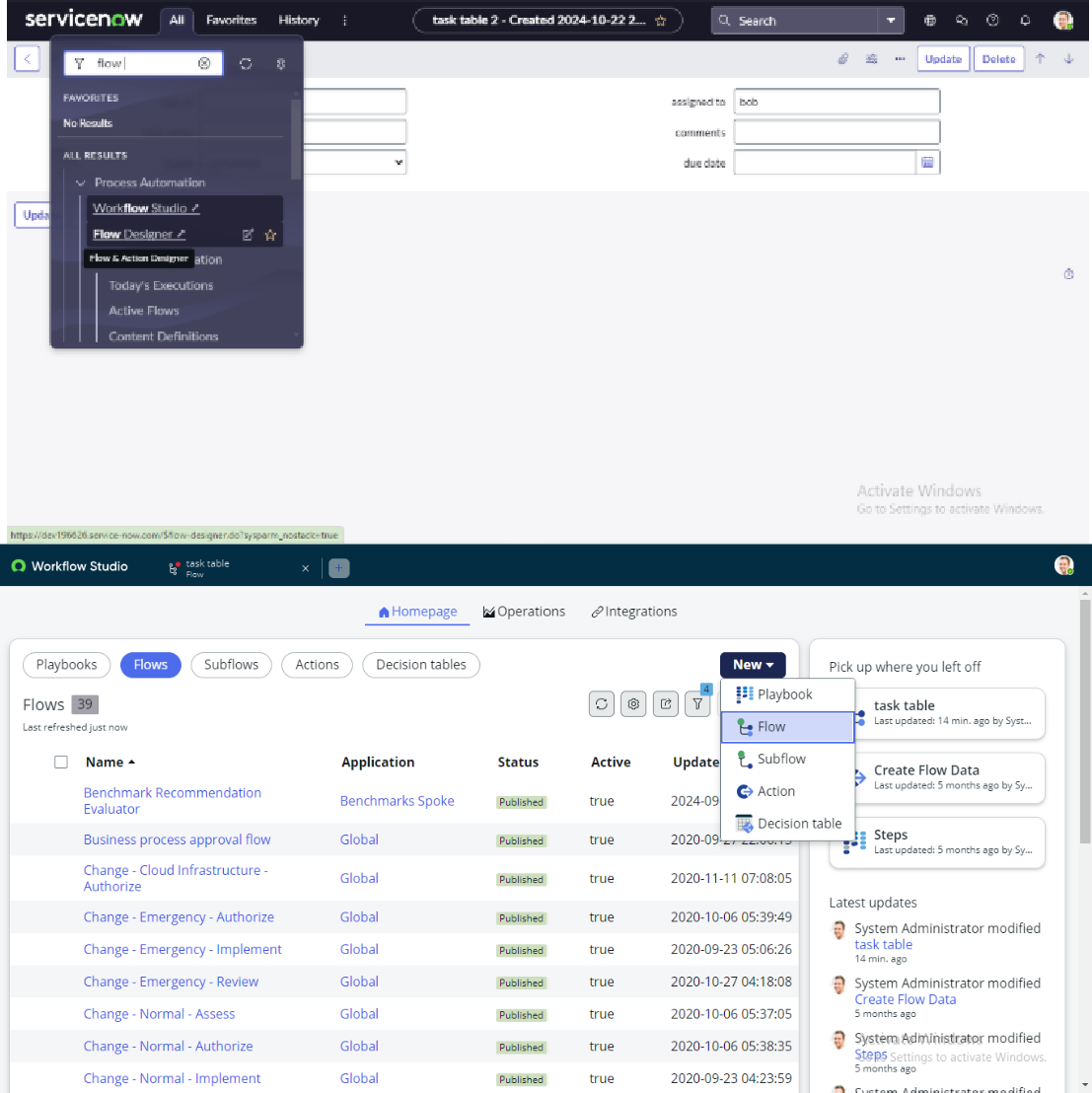
******

***Phase 6 :Assign table access to application***

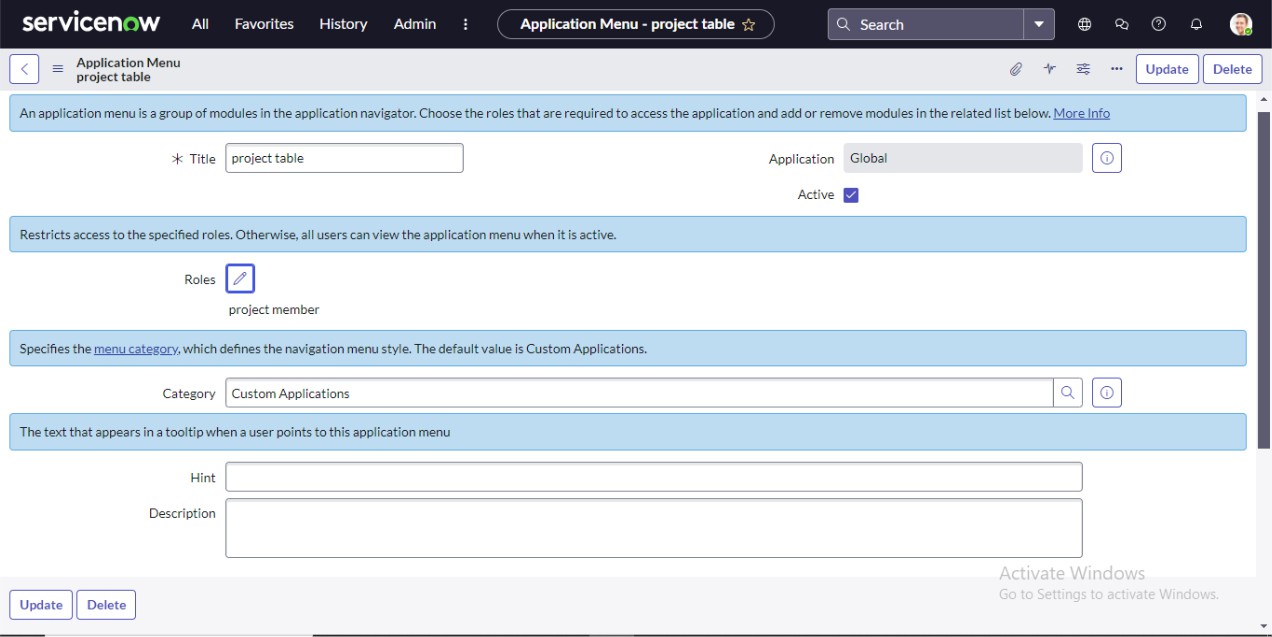
******

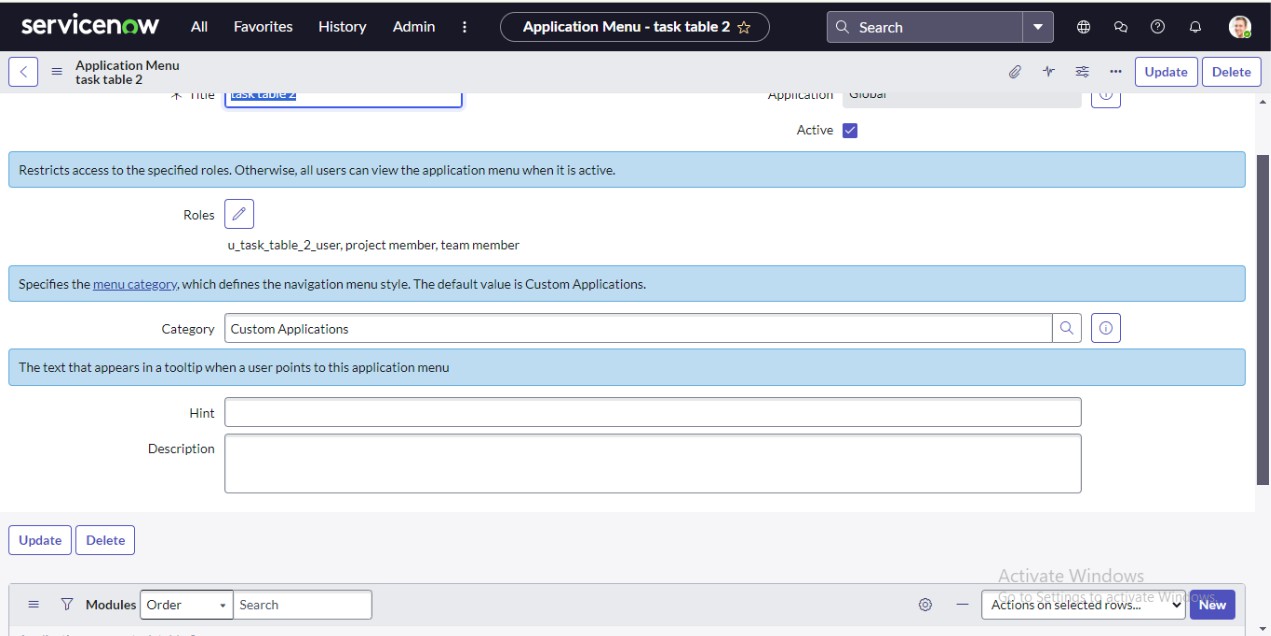
******

***Phase 8 :Create a Flow to Assign operations ticket to group***

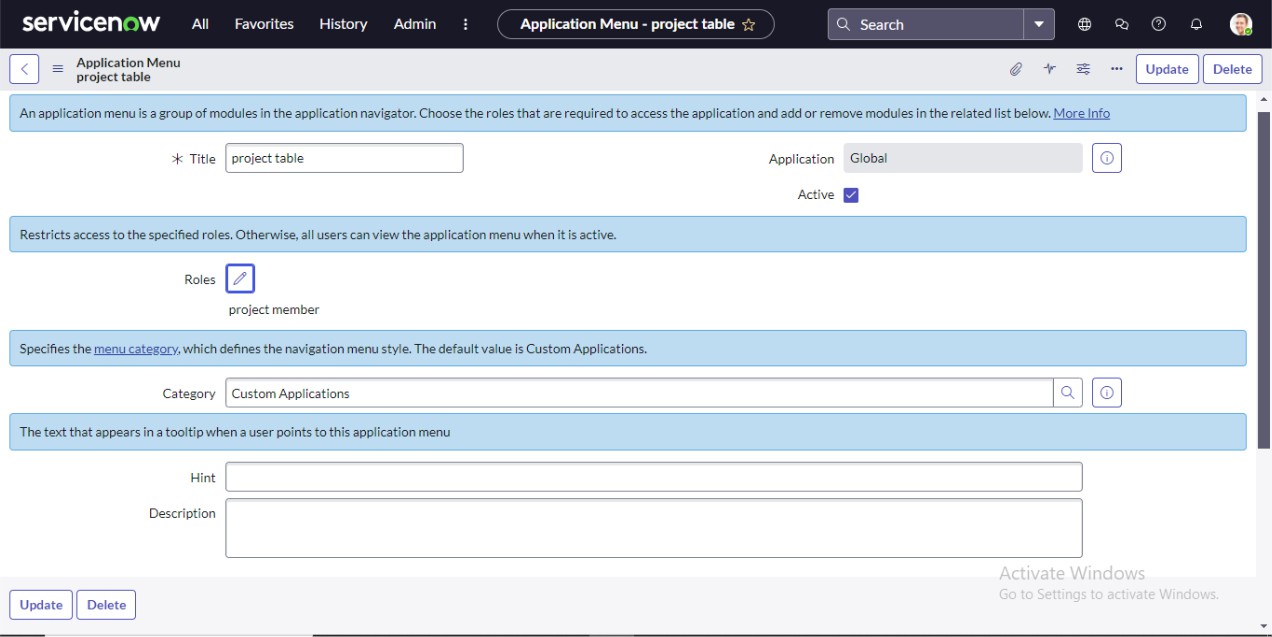
******

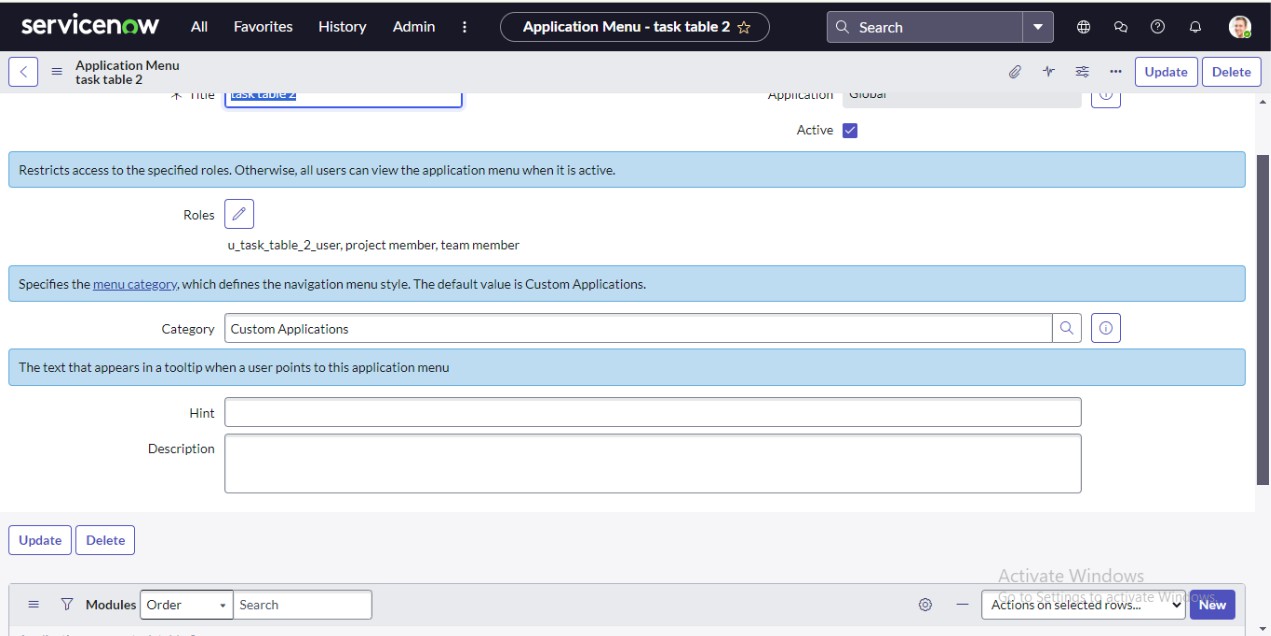
***Phase 6 :Assign table access to application***

******

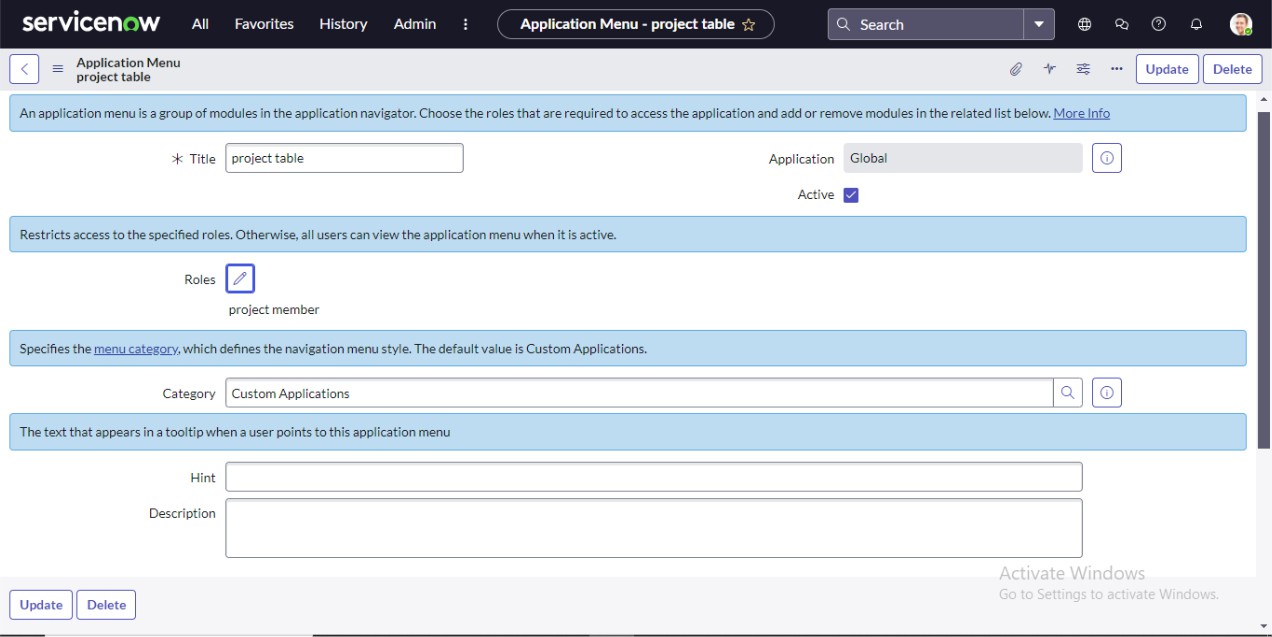
******

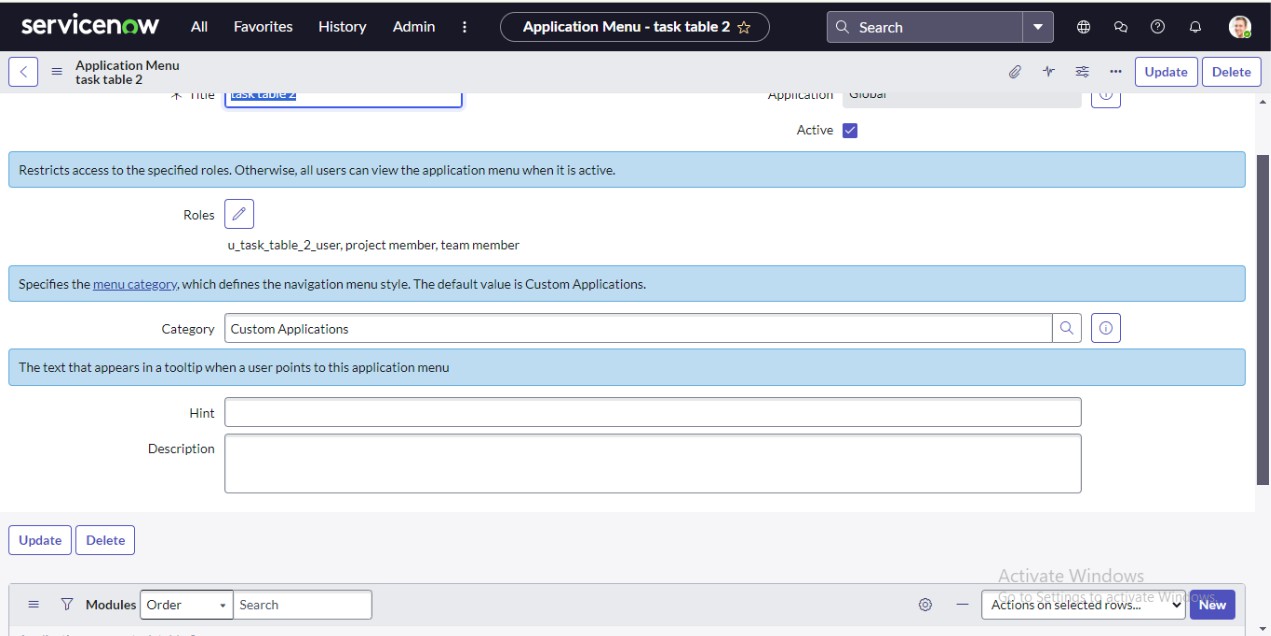
***Phase 6 :Assign table access to application***

******

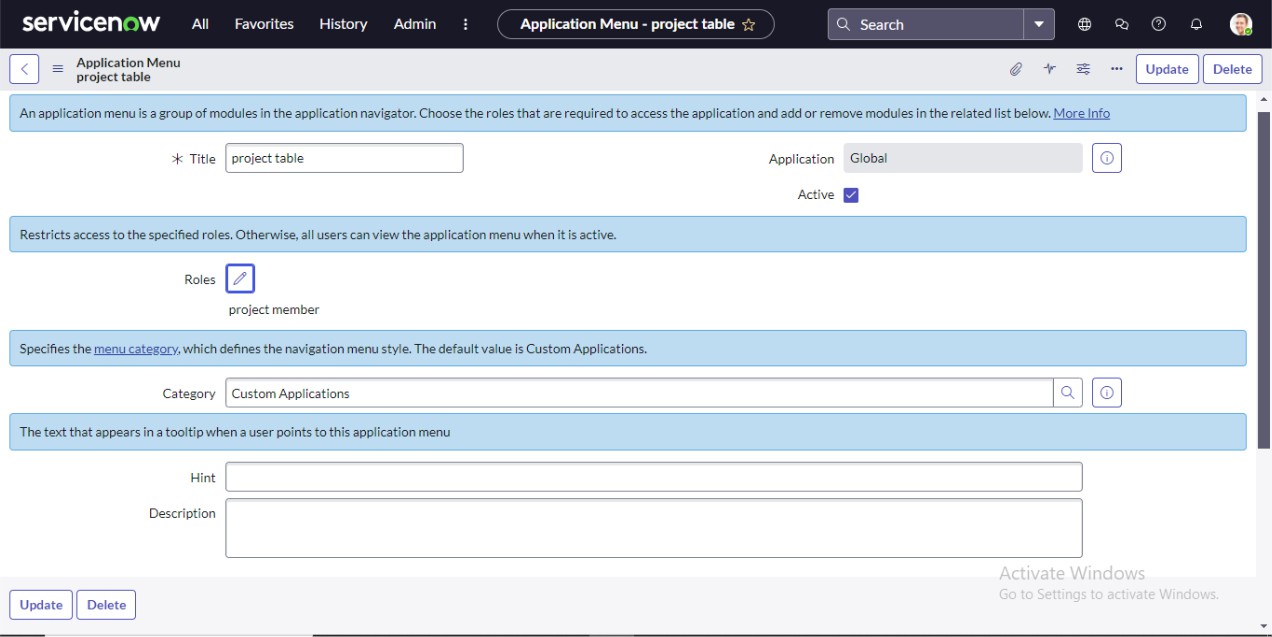
******

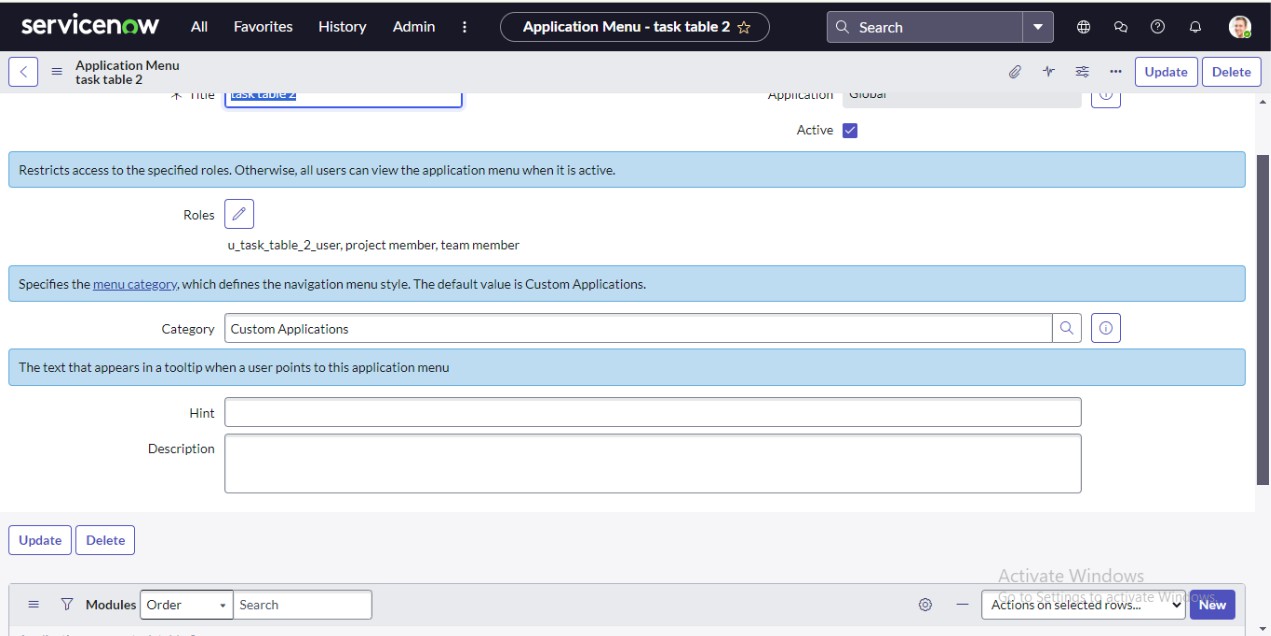
***Phase 6 :Assign table access to application***

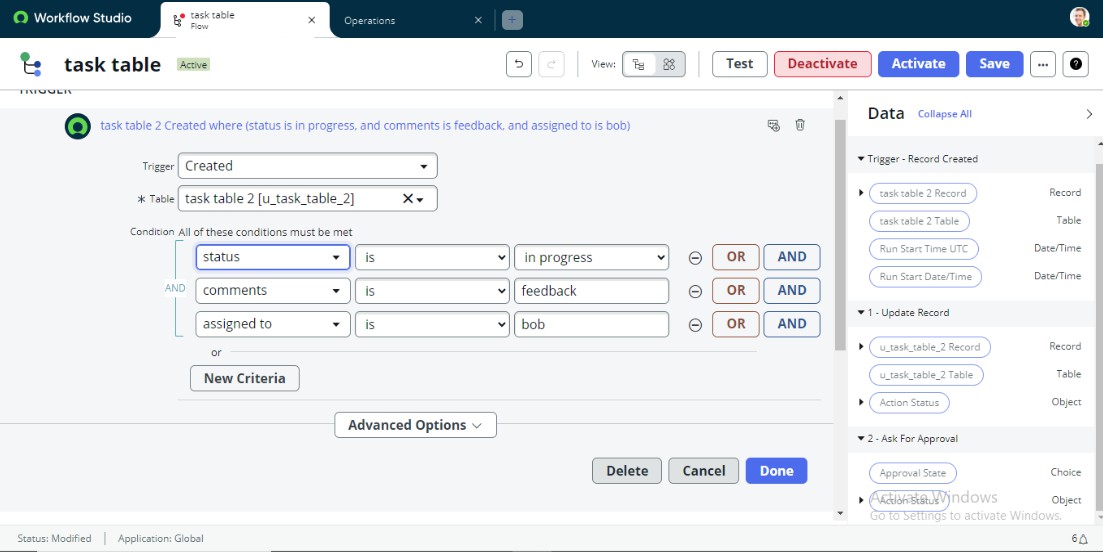
******

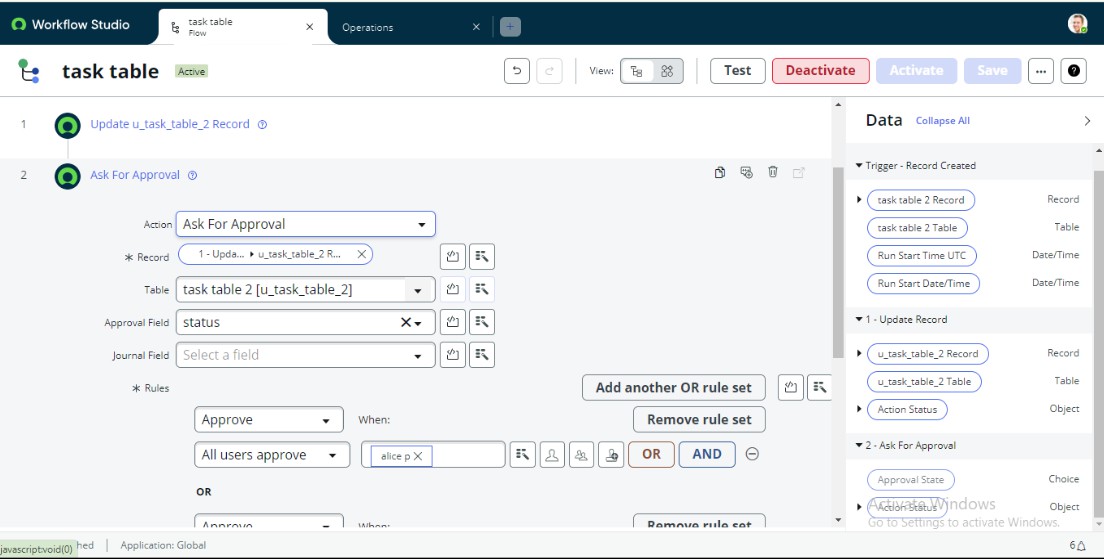
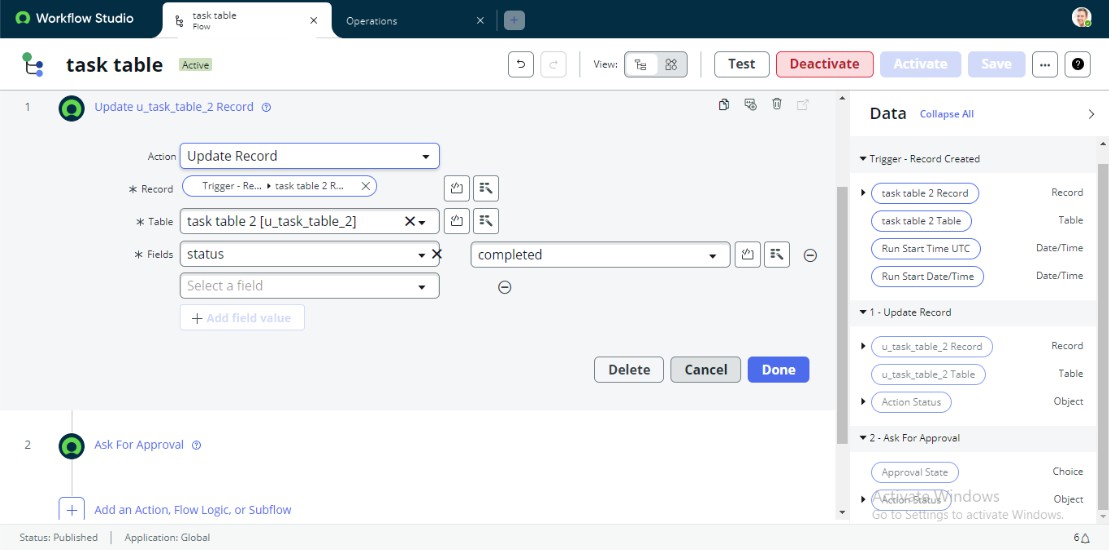
******

***Phase 6 :Assign table access to application***

******

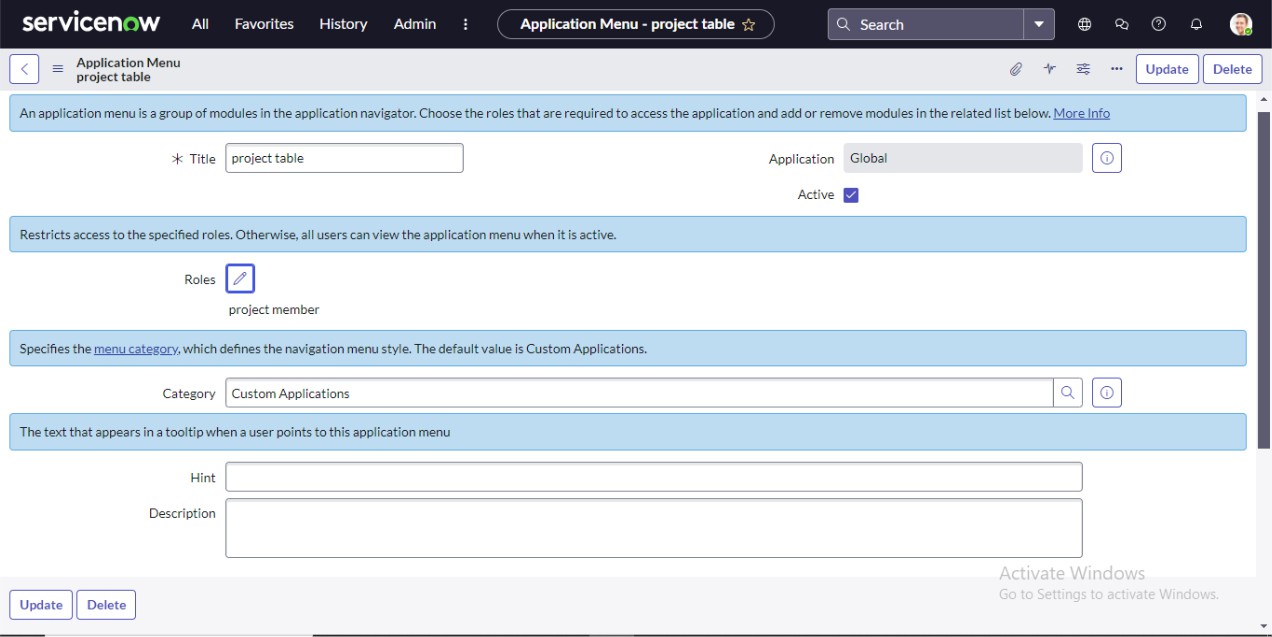
******

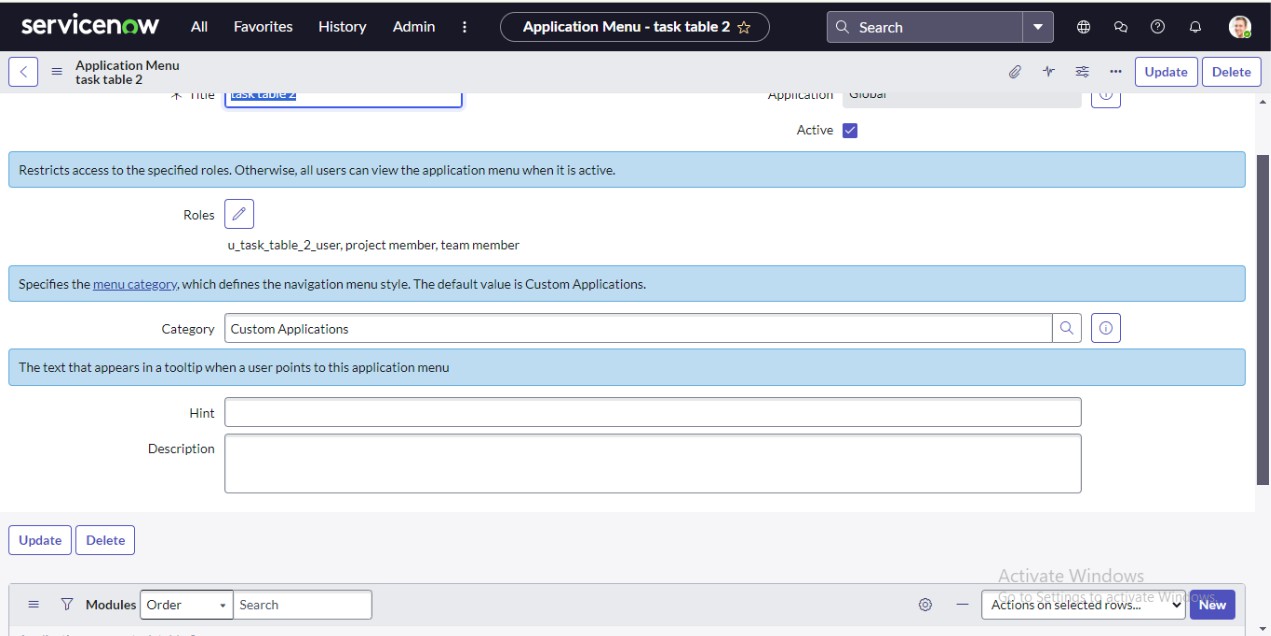


******

28

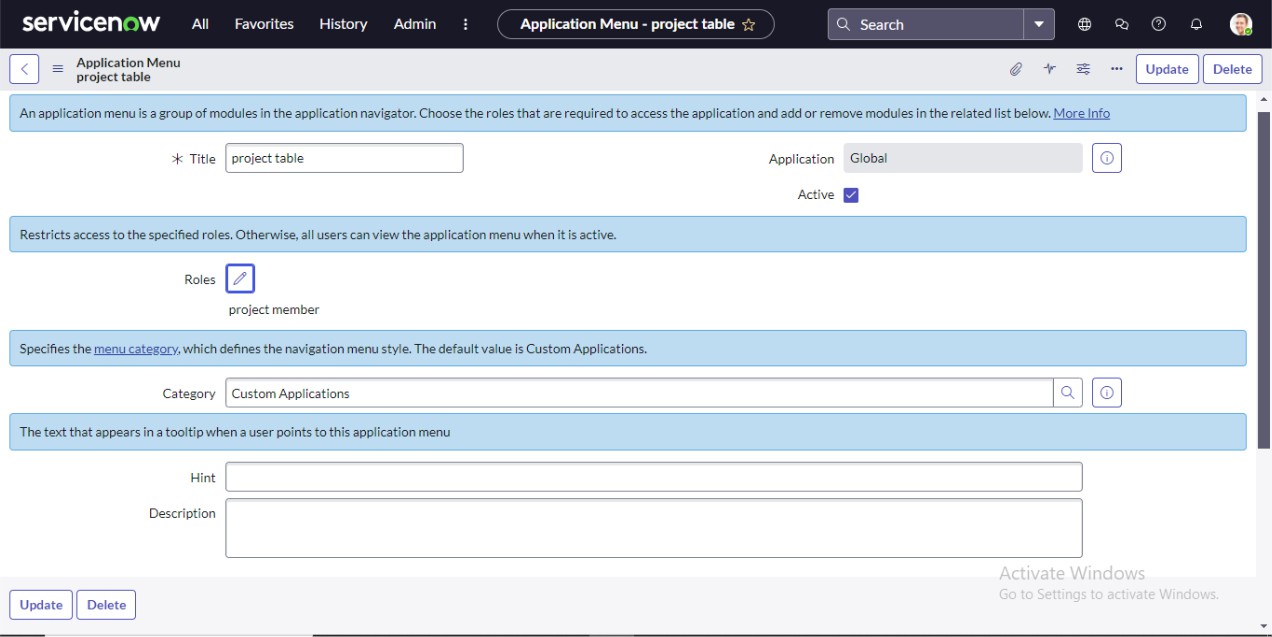
***Phase 6 :Assign table access to application***

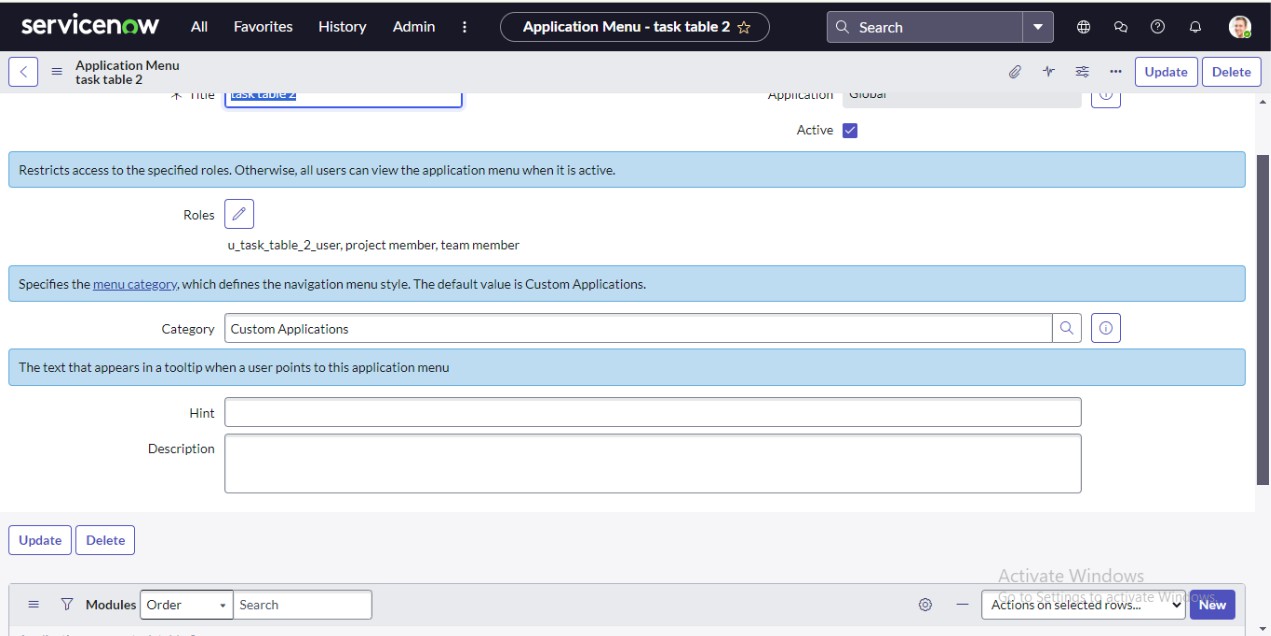
******

******

29

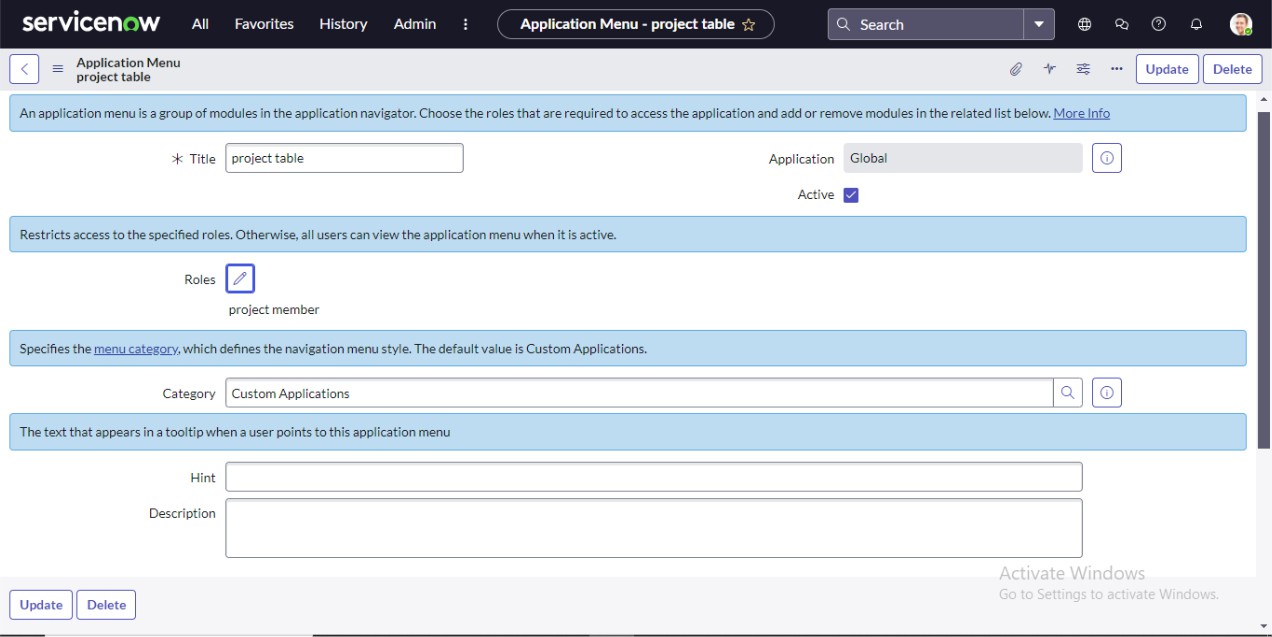
***Phase 6 :Assign table access to application***

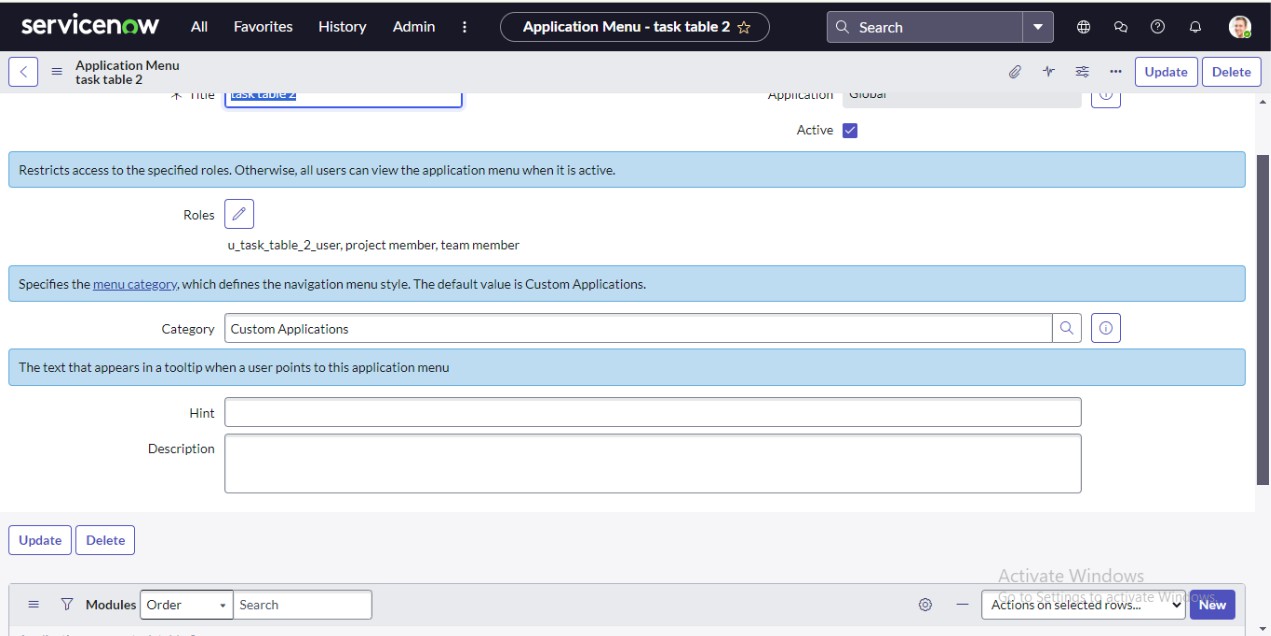
******

******

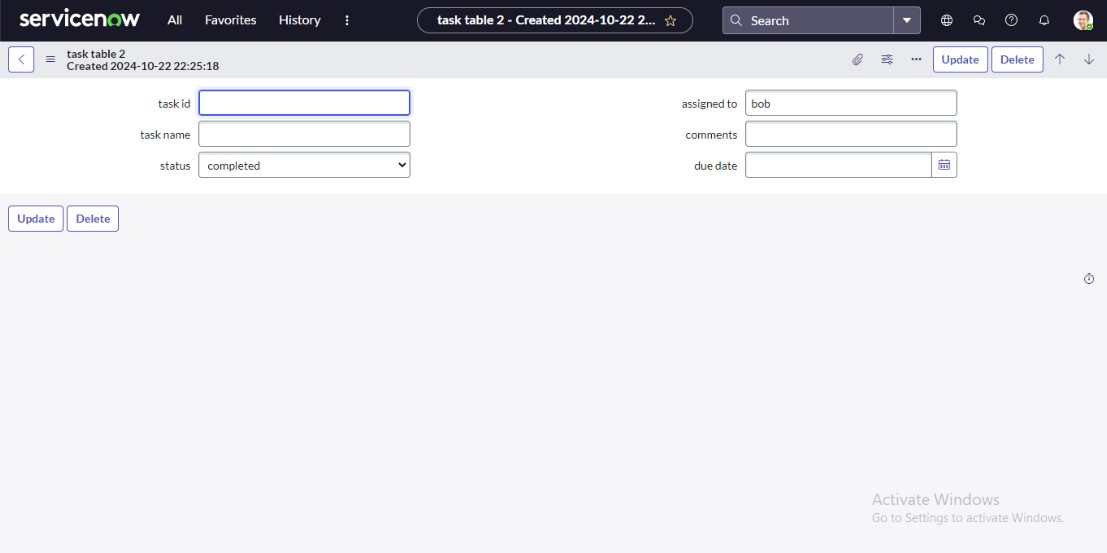
30

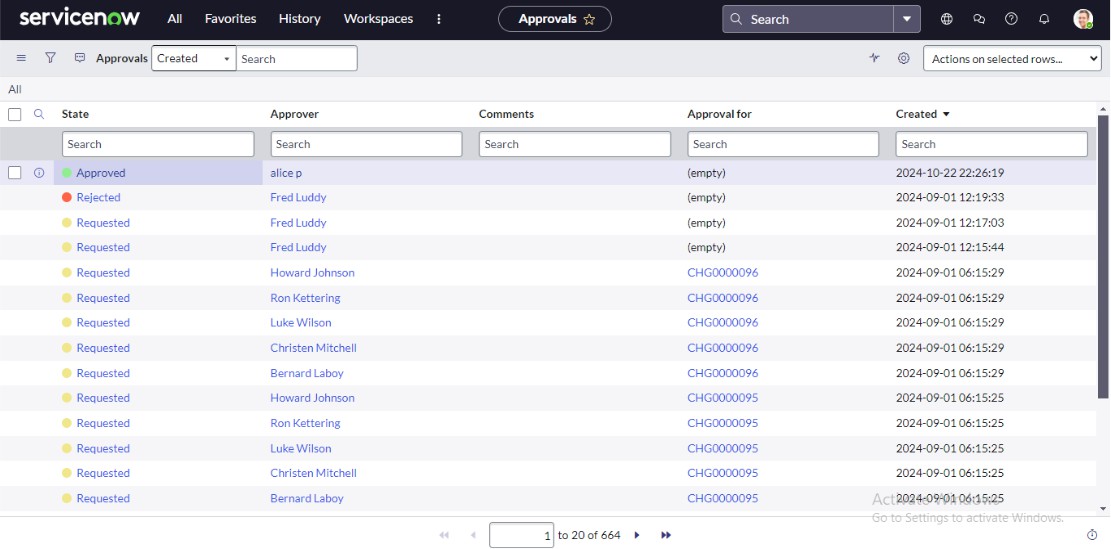
***Phase 6 :Assign table access to application***

******

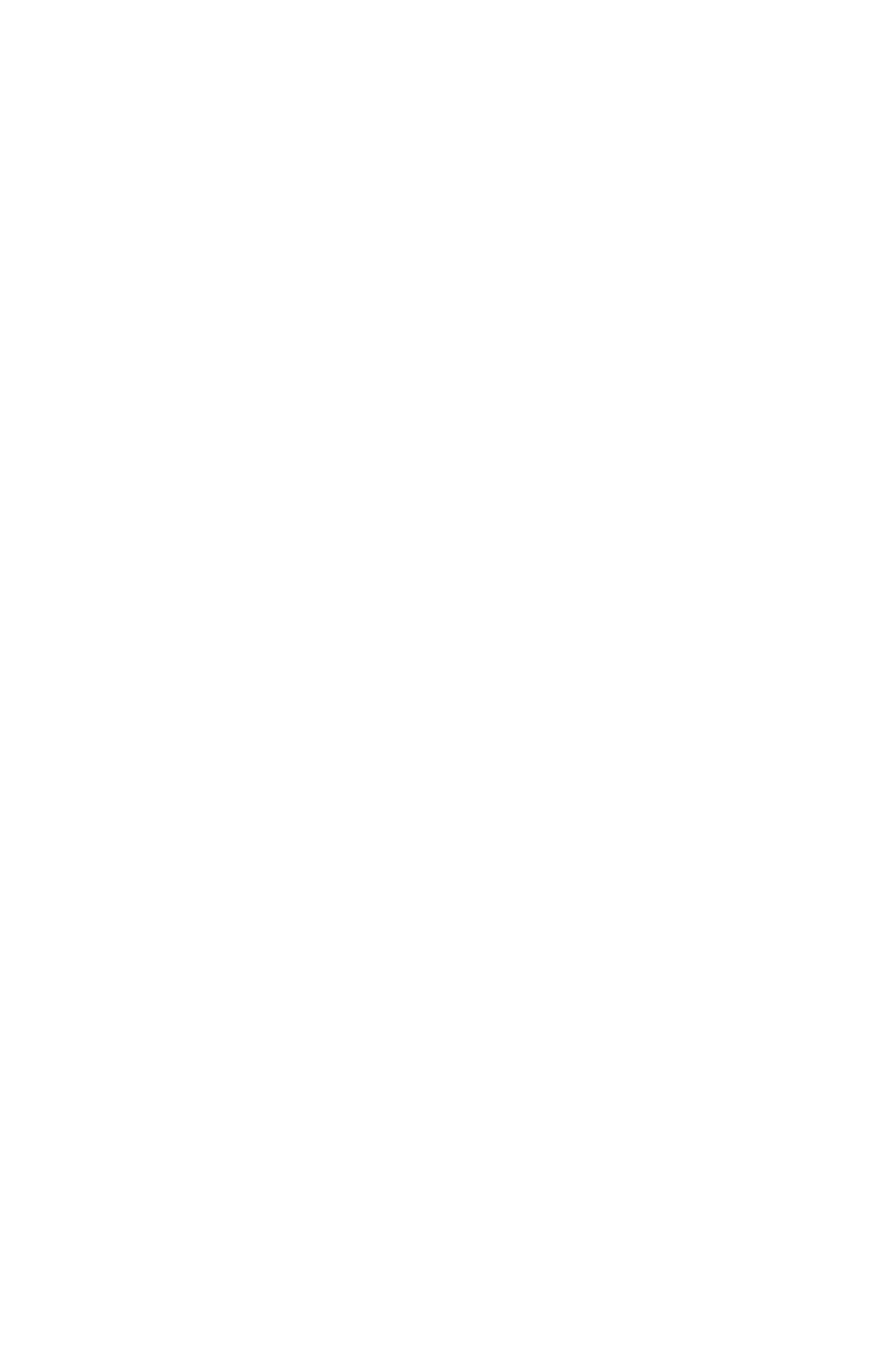
******

31





32



# PROJECT DESIGN PHASE

33

## Problem – Solution fit

**Problem**

In the current project management setup, a small team consisting of a Project Manager (Alice) and a Team Member (Bob) struggles to manage tasks effectively due to the absence of a structured workflow. The lack of role- based access control, clear responsibilities, and progress-tracking mechanisms leads to confusion, redundant work, and delayed project completion.

Communication gaps further make it difficult to monitor ongoing tasks and maintain accountability.

**Proposed Solution**

The proposed solution is a Role-Based Project Management System (RB-PMS) that streamlines task management, defines user roles, and introduces access control mechanisms.

This system allows the Project Manager to assign and monitor tasks while restricting the Team Member to viewing and updating only their assigned work. Real-time tracking, automated notifications, and progress dashboards will ensure transparency and accountability throughout the project lifecycle.

The system aims to:

 Reduce confusion in task allocation.

 Improve transparency and communication.

 Increase accountability through defined access rights.  Enhance project efficiency with structured workflows.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Conclusion

Parameter User Roles

Task Assignment Progress Tracking Access Control Notifications Activity Log

Reports

Description

Defines specific access levels for Project Manager and Team Member.

Allows creation, allocation, and prioritization of project tasks.

Displays real-time task status and completion percentage.

Restricts data modification based on user role.

Sends automatic alerts for updates, deadlines, or task changes.

Records all user actions for accountability and audit purposes.

Generates performance and progress summaries for management.

The proposed role-based project management system effectively resolves issues of unclear responsibilities and poor task tracking in small teams. By defining distinct roles for the Project Manager and Team Member, the system ensures accountability, transparency, and efficient workflow management. Features such as task assignment, progress tracking, and notifications enhance coordination and productivity. Although designed for small teams, the system provides a solid foundation for future scalability with advanced features like AI-based task automation and cloud integration, making it a reliable solution for improving project efficiency and team collaboration.

### Solution Architeture

**Goal of the Architecture**

The main goal of the proposed Role-Based Project Management System (RB-PMS) architecture is to create a secure, scalable, and efficient framework that facilitates task management, user accountability, and progress tracking.

The architecture is designed to:

 Define clear user roles and access levels.

 Streamline task assignment and monitoring processes.

 Enable real-time collaboration and communication between team members.  Ensure data integrity, security, and scalability for future expansion.

### Key Components of the Architecture

1. User Interface (UI) Layer:

 Provides a simple and user-friendly interface for both Project Manager and Team Member.  Supports both web and mobile platforms for accessibility.

 Displays dashboards, task lists, and notifications in real time.

1. Application Layer:

 Acts as the core of the system where business logic and workflows are implemented.

 Includes modules for role-based authentication, task management, and notification control.  Ensures that only authorized actions are performed based on user roles.

1. Database Layer:

 Stores all critical data including user credentials, task information, progress updates, and activity logs.

 Ensures data consistency, confidentiality, and quick retrieval through structured queries.

1. Notification and Reporting Layer:

 Generates automated alerts for new tasks, updates, and deadlines.

 Provides analytical dashboards and downloadable reports for project tracking and evaluation.

Development Phases

### Phase Description

**Phase 1:** Requirement Analysis Identify user needs, system goals, and workflow challenges between Project Manager and Team Member.

**Phase 2:** System Design Create system architecture diagrams, define database schema, and finalize UI wireframes.

**Phase 3**: Development Implement modules for authentication, task assignment, and progress tracking using chosen technologies.

**Phase 4:** Testing and Validation Perform functional, usability, and security testing to ensure reliability and role-based access control.

P**hase 5:** Deployment Deploy the system on a local or cloud environment, ensuring accessibility for both users.

**Phase 6:** Maintenance C Enhancement Monitor performance, fix bugs, and integrate advanced features like AI-based task prediction and analytics.

### Solution Architecture Description

.

### User Interface (UI):

The front-end interface allows the Project Manager and Team Member to interact with the system using a web or mobile application.

### Application Layer:

Handles the core logic of the system, including user authentication, task management, notifications, and access control. The role-based module ensures that each user performs only their permitted actions.

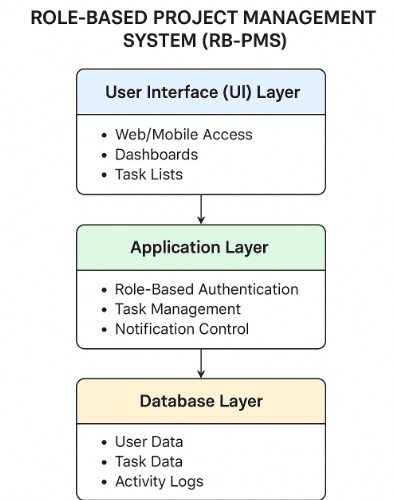
### Database Layer:

Stores user credentials, task details, progress status, and activity logs securely. It supports quick data retrieval and update operations.

### Reporting and Analytics Layer:

Provides graphical dashboards and analytical reports to visualize task completion rates, user performance, and overall project progress.

### Solution Architecture Diagram

****

37

# PROJECT PLANNING PHASE

38

Project Planning Phase 1. Product Backlog

Backlog ID

PB-01

PB-02

PB-03

PB-04

PB-05

Priority

High

High

Medium

Medium

Low

Feature / Task

User Authentication Module

Task Creation & Assignment

Progress Dashboard

Notifications & Alerts

Reports & Analytics

Description

Implement secure login and role- based access for Project Manager and Team Member.

Allow Project Manager to create, assign, and prioritize tasks.

Provide visual tracking of ongoing tasks with completion percentages.

Send automated notifications on updates, task completion, and deadlines.

Generate performance and progress reports for management insights.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Sprint Planning

Sprint No.

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Duration

Week 1–2

Week 3–4

Week 5–6

Week 7

Sprint Goal

Implement authentication and

Develop core task management

Enable notifications and

Testing and Deployment.

Key Deliverables

Login module, Role-based

Task creation, assignment, and

Notifications module, Report

Unit testing, bug fixing, system

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. User Stories

User Story ID

US-01

US-02

US-03

US-04

User Role

Project Manager

Team Member

Project Manager

Team Member

User Story

As a Project Manager, I want to assign tasks to my team so that responsibilities are clearly divided.

As a Team Member, I want to update task progress so that the manager can monitor my work.

As a Project Manager, I want to generate reports to review overall project performance.

As a Team Member, I want to receive notifications for new or updated tasks.

Acceptance Criteria

The system should allow creation and assignment of tasks with deadlines.

Updates must reflect immediately in the manager’s dashboard.

The system should generate downloadable reports summarizing task status.

Alerts should be sent automatically for assigned or modified tasks.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Story Points

Story ID

SP-01 SP-02 SP-03

Feature / Functionality

Login & Role Setup

Task Assignment Module

Progress Dashboard

Effort Level

Medium High Medium

Story Points

5

8

6

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# REQUIREMENT ANALYSIS

27

1. Solution Requirements

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

42

Type

Functional Requirements

FR1

FR2

FR3

FR4

FR5

FR6

Non-Functional Requirements

NFR1

NFR2

NFR3

NFR4

NFR5

Requirement Description

The system must allow the Project Manager to log in securely using unique credentials.

The Project Manager should be able to create, assign, and prioritize project tasks.

Team Members must be able to view their assigned tasks and update progress.

The system should generate project progress reports for the manager.

Notifications should be sent automatically for new assignments or updates.

The dashboard should display real-time task status and completion progress.

The system should have a user-friendly and responsive interface.

Data should be stored securely with proper authentication and access control.

The application should support concurrent access by multiple users.

The system should ensure 99% uptime and quick response (<2 seconds) for all operations.

The system must allow scalability for future feature expansion.

1. Data Flow Diagram (DFD – Level 1) Description:
2. Start:

 The system initiates when a user (either Project Manager or Team Member) opens the application.

1. User Login / Authentication:

 The user enters credentials (username and password).  The system verifies login details from the database.

 If valid, the user is redirected to their respective dashboard (Manager or Member).  If invalid, an error message is displayed.

1. Role Identification:

 Project Manager: Gets access to task creation, assignment, and report generation.  Team Member: Gets access to view assigned tasks and update progress.

4 .Task Management:

 The Manager creates tasks with details (title, description, due date, and assignee).  Tasks are stored in the database and linked to assigned members.

1. Progress Update:

 Team Members mark progress (e.g., Not Started → In Progress → Completed).  Updates are reflected in real time on the Manager’s dashboard.

1. Notification Trigger:

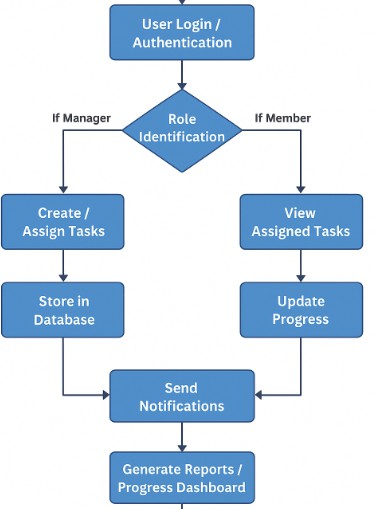
 When a new task is created or updated, the system automatically triggers a notification.  Notifications are sent to concerned users (via email, app alert, or message).

1. Report Generation:

 The Project Manager generates reports to analyze progress and performance.  Reports summarize tasks by status, completion rate, and deadlines.

1. End:

 The process continues in cycles until the project is completed or archived



43

1. Technology Stack

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

44

Layer

Frontend

Backend

Database

APIs

Authentication

Notifications

Hosting / Deployment

Version Control

Technology Used

HTML, CSS, JavaScript (React or Angular optional)

Node.js / Java (Spring Boot) / Python (Flask or Django)

MySQL / MongoDB

RESTful APIs

JWT (JSON Web Token) / OAuth2

Firebase Cloud Messaging / Email Service

AWS / Google Cloud / Local Server

GitHub / GitLab

Purpose / Description

For building an interactive and responsive user interface.

Handles authentication, business logic, and server-side processing.

Stores user credentials, task details, and progress data.

Enables communication between frontend and backend.

Ensures secure user login and access control.

Sends task updates and alerts.

Provides reliable and scalable deployment.

For collaborative development and source code management.

## GENERATIVE AI IN ACTION

Artificial intelligence has evolved through numerous phases, yet it remains both

intriguing and inspiring to witness machines become increasingly capable of crafting poetry, humor, and responses that uncannily mimic human creativity. You’ll learn about the history of AI, how deep learning plays a pivotal role in generative AI (“gen AI”), and how gen-AI

works and is applied to various industries. Throughout this course, you will also learn how to create algorithms, and gain hands-on experience writing code using popular programming languages.

Completed the following required modules to earn an industry-recognized IBM SkillsBuild digital credential called **Generative AI in Action:**

* 1. Introduction to Generative AI
  2. Crafting Precision Prompts with Generative AI
  3. Coding Simplified with Generative AI

After completing Generative AI in Action, I was able to:

* + - Explain how generative artificial intelligence works
    - Define what foundation models are and their role in machine learning
    - Understand how transformers models are used to solve various language-related tasks
    - Describe how prompt engineering improves generative AI models
    - Identify common prompt elements
    - Explore code generation using a natural language prompt
    - Describe prompt engineering techniques

Perform common programming tasks using Python’s built-in functions and libraries

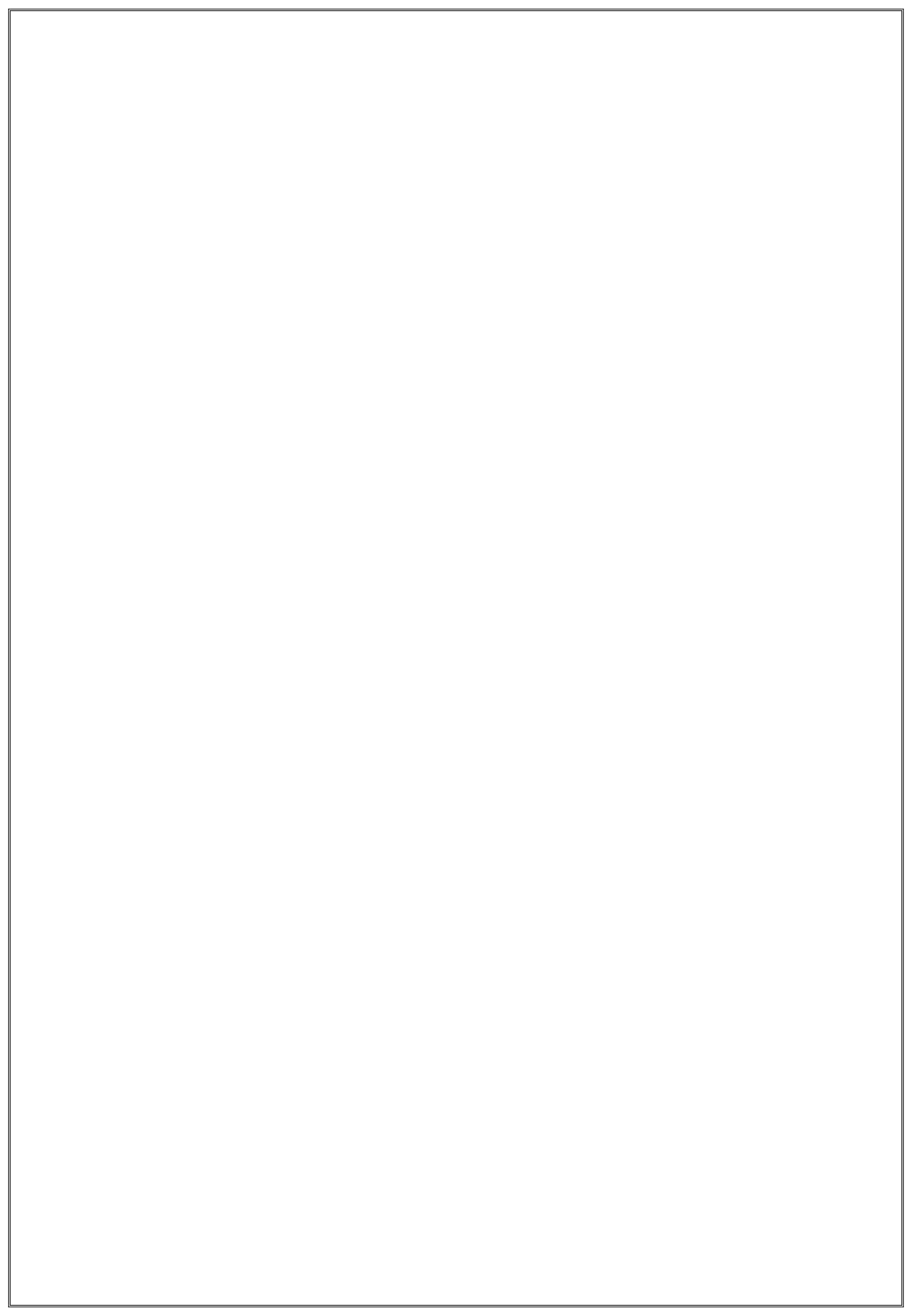
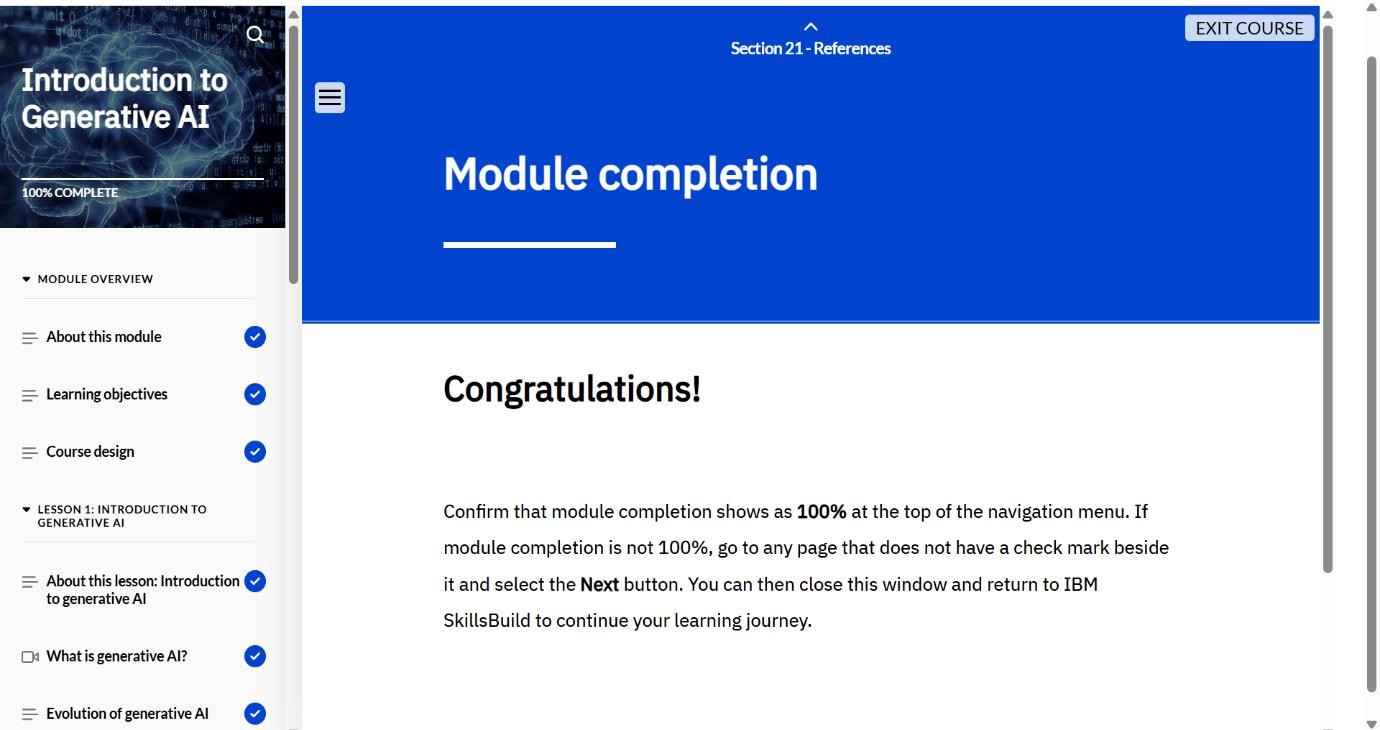
Create scripts and code for solving real-world problems and automating routine tasks

## MODULE 1 - Introduction to Generative AI

In this course, I have learned about the history of AI,how deep learning plays a pivotal role in generative AI (gen-AI), and how gen-AI works andis applied to various industries. I also learned how to create algorithms, and gain hands-onexperience writing code using popular programming languages.

After completing this module, I was able to:

* + - Explain how generative artificial intelligence (gen-AI) works
    - Define what foundation models are and their role in machine learning
    - Understand how transformers models are used to solve various language-related tasks



* + - Describe how prompt engineering improve generative AI models
    - Perform common programming tasks using Python's built-in functions and libraries
    - Create scripts and code for solving real-world problems and automating routine tasks

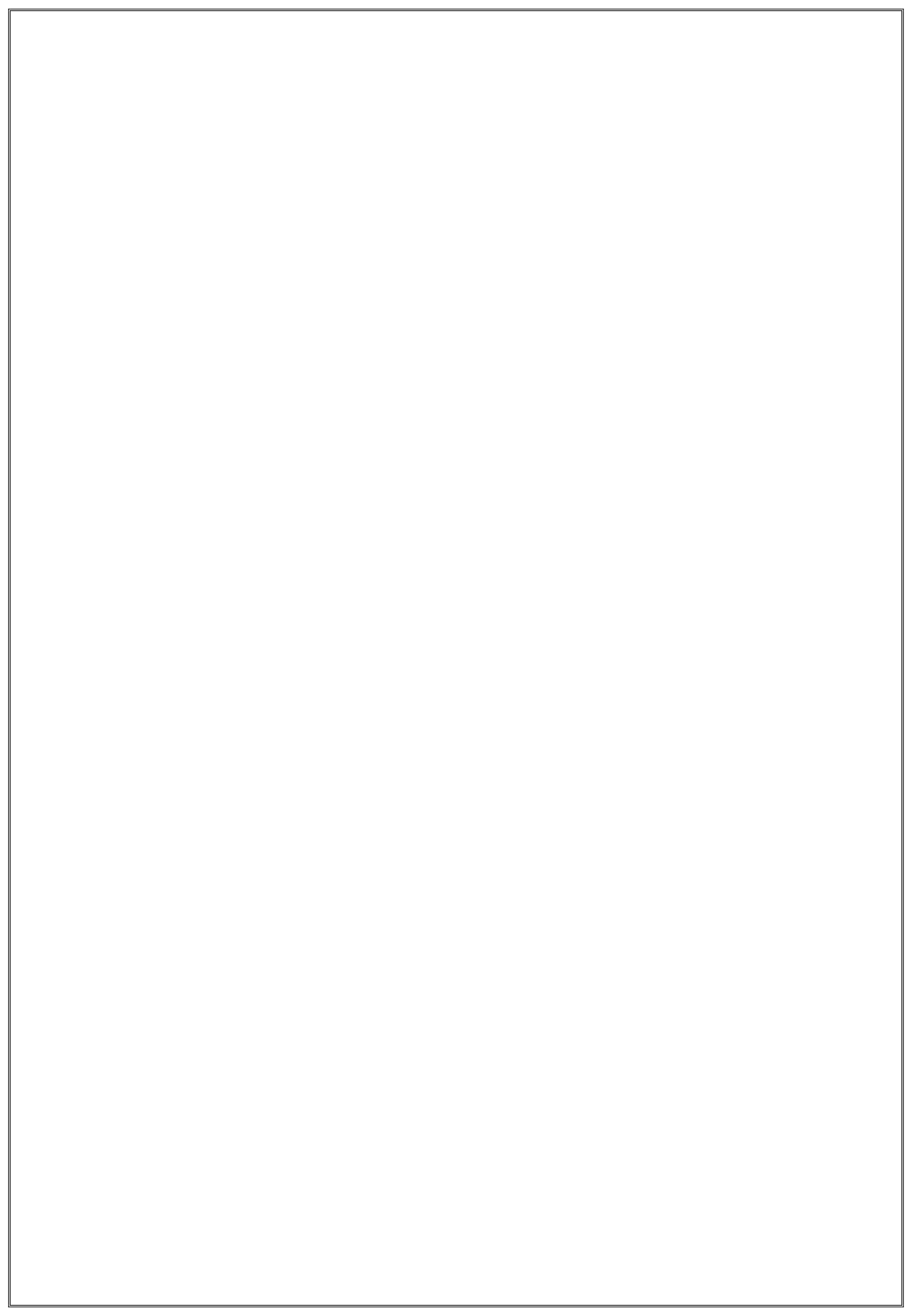
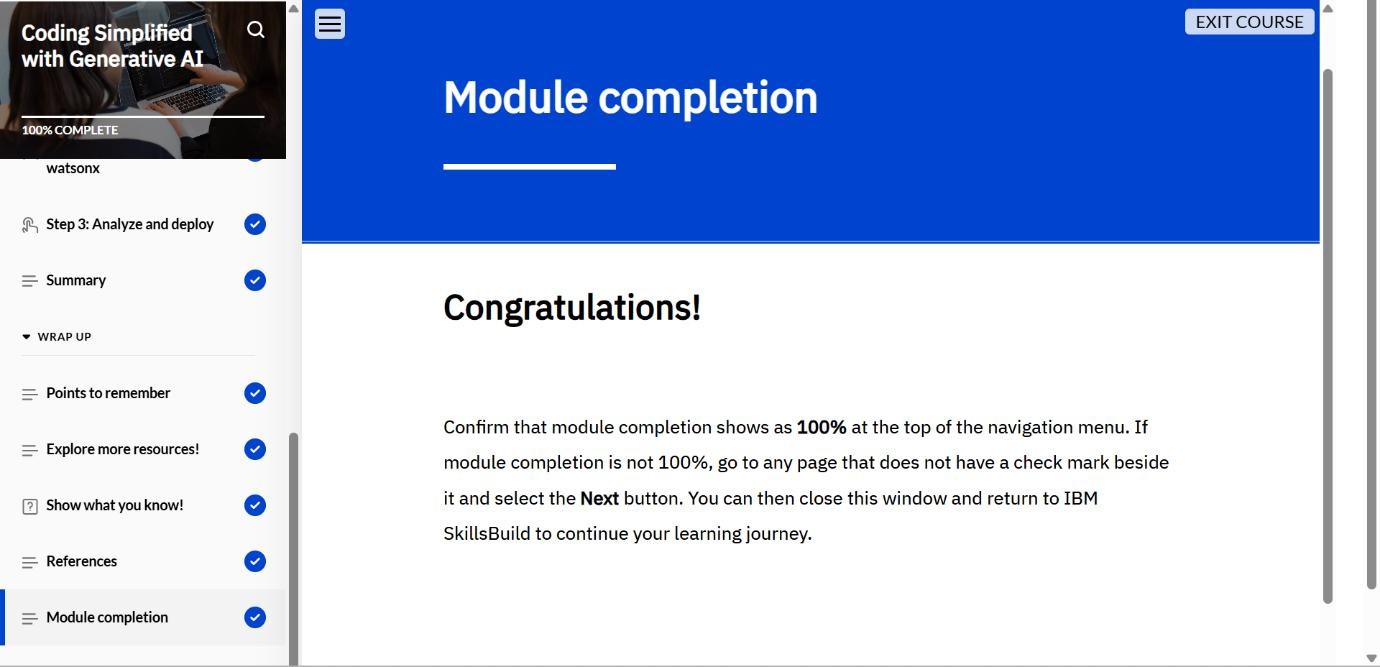
## MODULE 2- Crafting Precision Prompts with Generative AI

This was an activity-based course. I learned about AI language models and the rules

to follow when giving instructions, or prompting, an AI language model. I walked-through a guided activity that demonstrates how to write effective prompts for an AI language model to help plan a travel itinerary. Finally, I participated in an activity to apply what I learned to effectively write prompts for an AI language model to create my own custom music playlist.

After completing this course, I was able to:

* + - Describe an AI language model
    - Explain how an AI language model understands and responds to humans
    - Identify the rules to follow to write effective prompts to generate focused and accurate results from an AI language model
    - List the steps to sign up for a ChatGPT account
    - Follow the steps to effectively write and refine a series of prompts for ChatGPT for a travel itinerary scenario
    - Demonstrate the steps to effectively write and refine a series of prompts for ChatGPT to create a custom music playlist



## MODULE 3- Coding Simplified with Generative AI

In this course, I had learned the basics of scripting, understand its distinctions from traditional programming, and explore how generative AI models are used to simplify and streamline code generation. Through hands-on labs, I also learned how to create algorithms and apply my skills using widely used programming languages.

After completing this module, I was able to:

* + - Define scripting and how it is works
    - Explain the differences between scripting and traditional programming and when each approach is used
    - Describe how Python is used to perform various tasks
    - Create a working Python application using IBM watsonx Code Generation

## CONCLUSION

The Generative AI in Action program provided a solid foundation in understanding the core principles and practical applications of generative artificial intelligence. Through the three modules—Introduction to Generative AI, Crafting Precision Prompts with Generative AI, and Coding Simplified with Generative AI—I developed a deep understanding of how AI models such as foundation and transformer models function, and how prompt engineering enhances their effectiveness. The hands-on activities improved my ability to craft accurate prompts, automate coding tasks, and generate creative as well as technical outputs using Python and AI-assisted tools. Overall, this program strengthened my knowledge of AI- driven innovation, improved my coding proficiency, and equipped me with essential skills to apply generative AI techniques in real-world scenarios across multiple industries.

## WELCOME TO SERVICE NOW

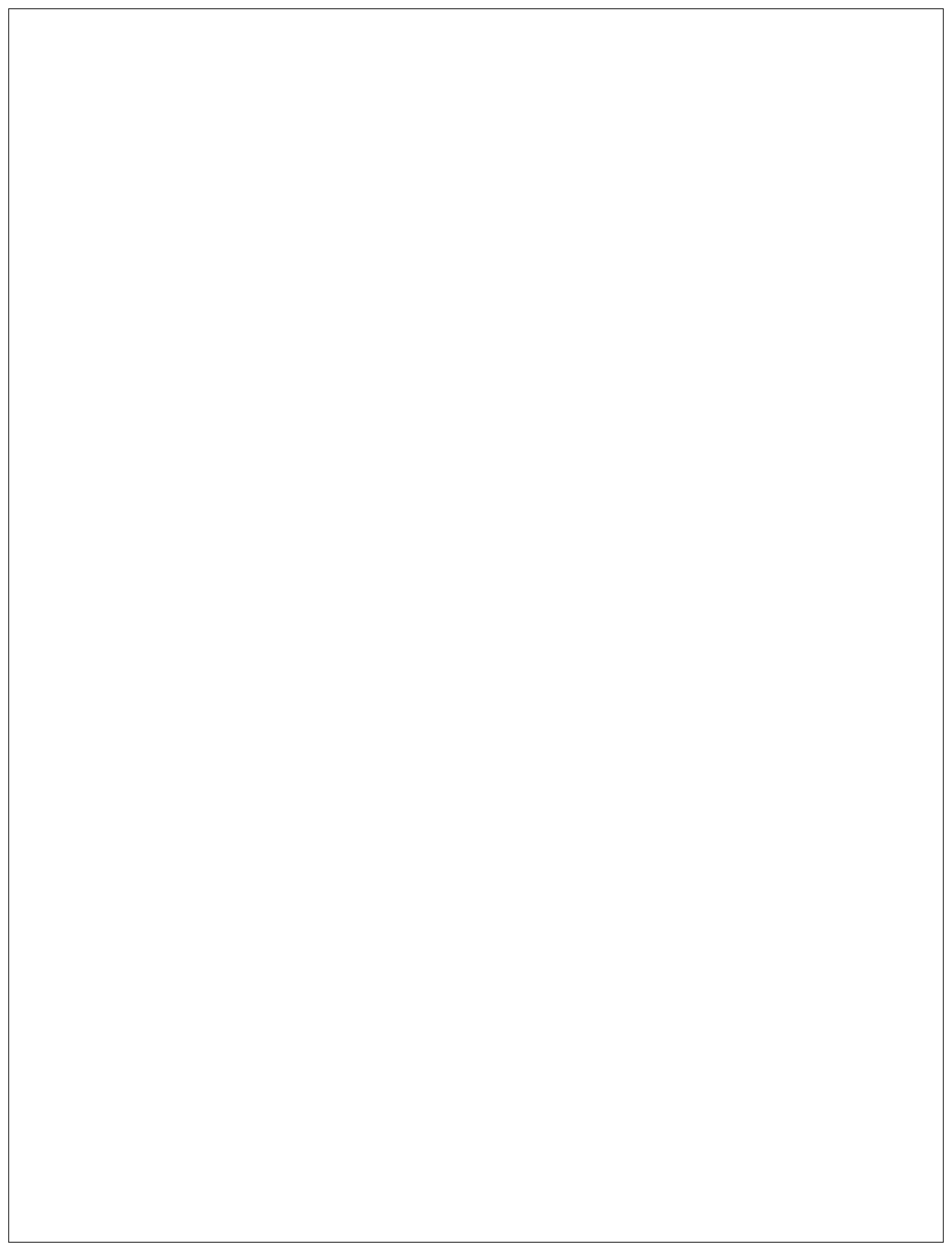
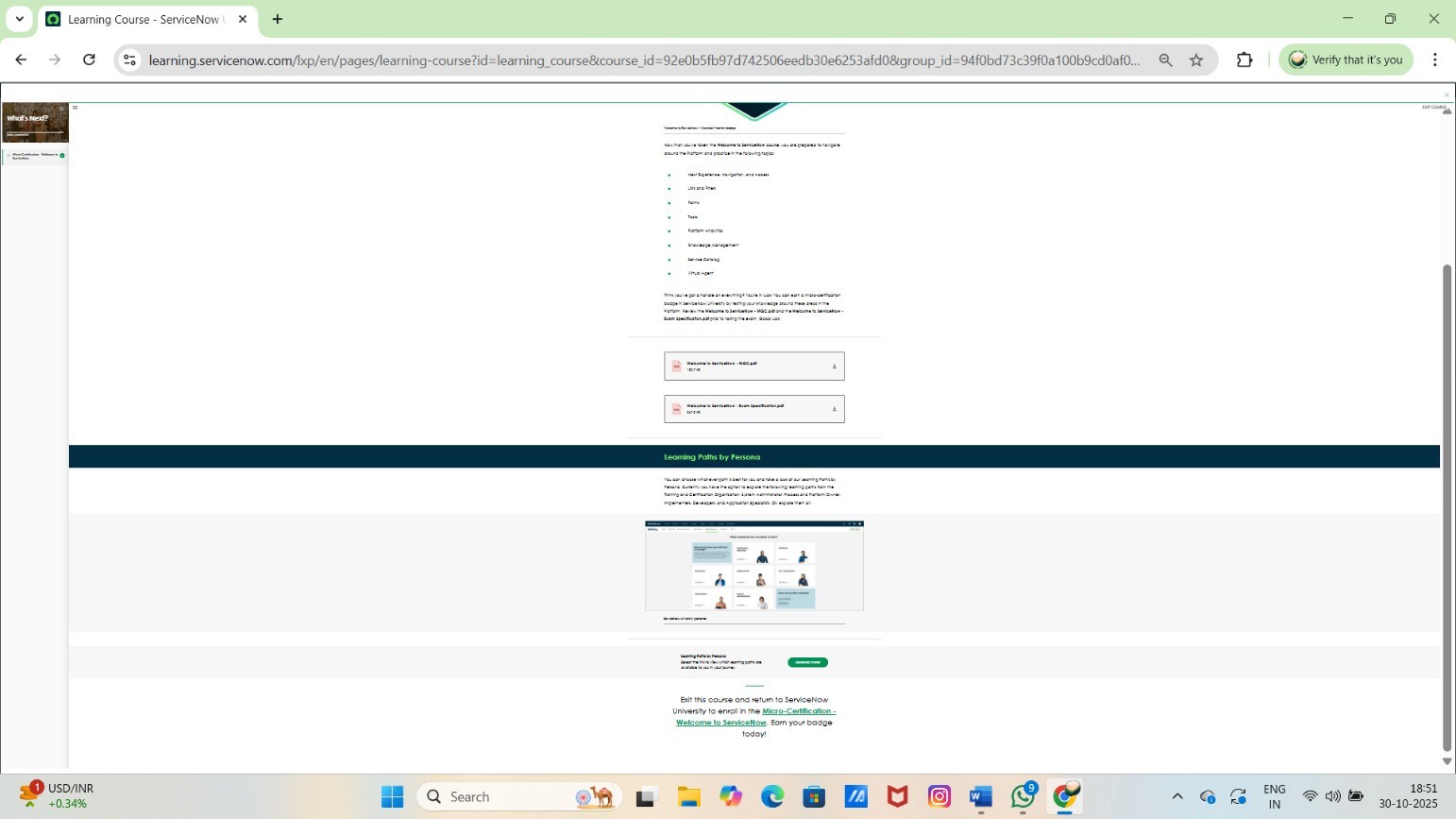
### Introduction to ServiceNow Platform

ServiceNow platform.

– Understand the purpose, features, and structure of

the

**Navigation and User Interface** – Learn how to use the interface, navigate between applications, and access key features.



**Core Modules and Tools** – Explore main modules like Lists, Forms, Tasks, Analytics, Knowledge Management, Service Catalog, and Virtual Agent.

**Practical Usage and Analytics** – Gain skills to manage data, perform platform analytics, and use ServiceNow effectively in real work environments.

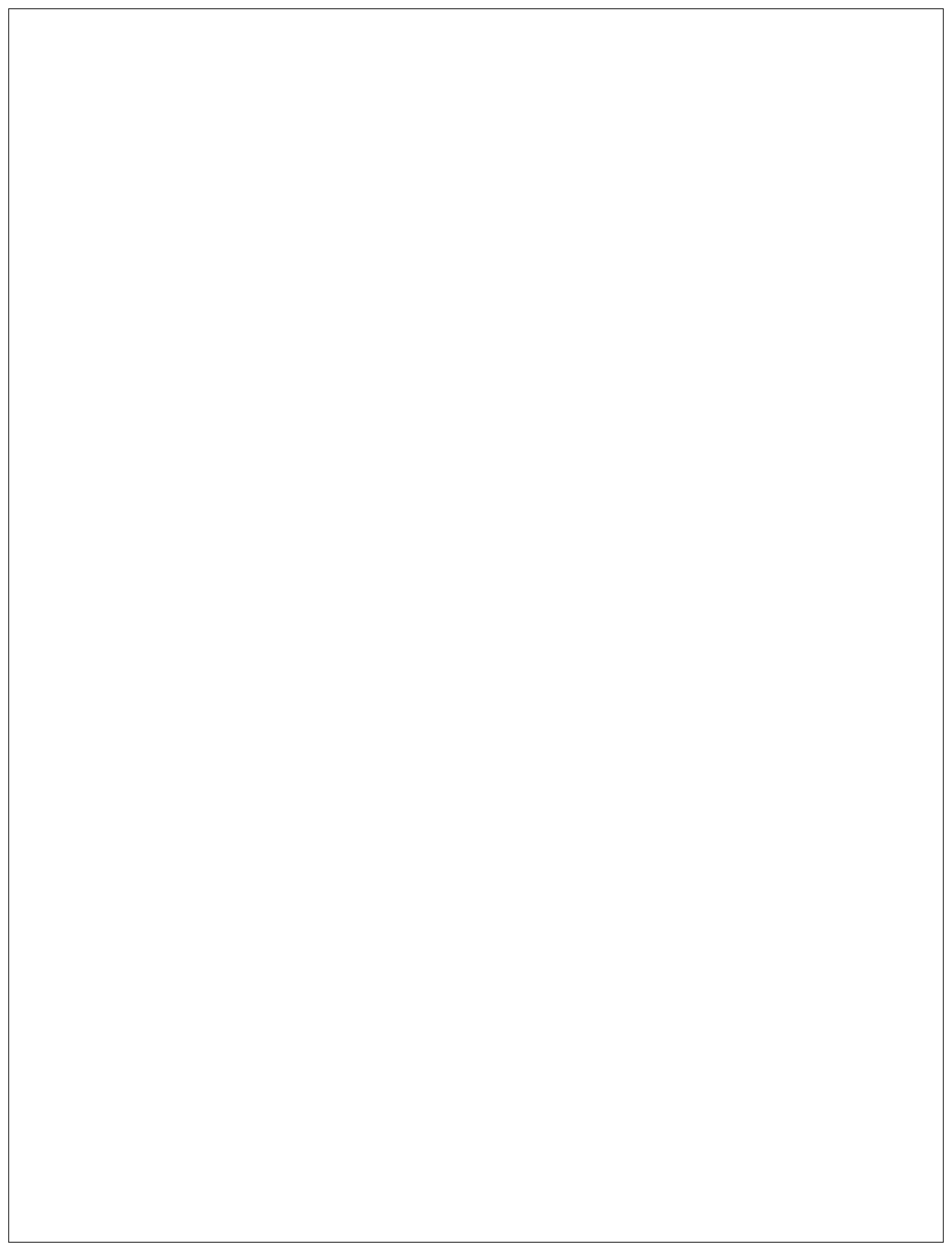
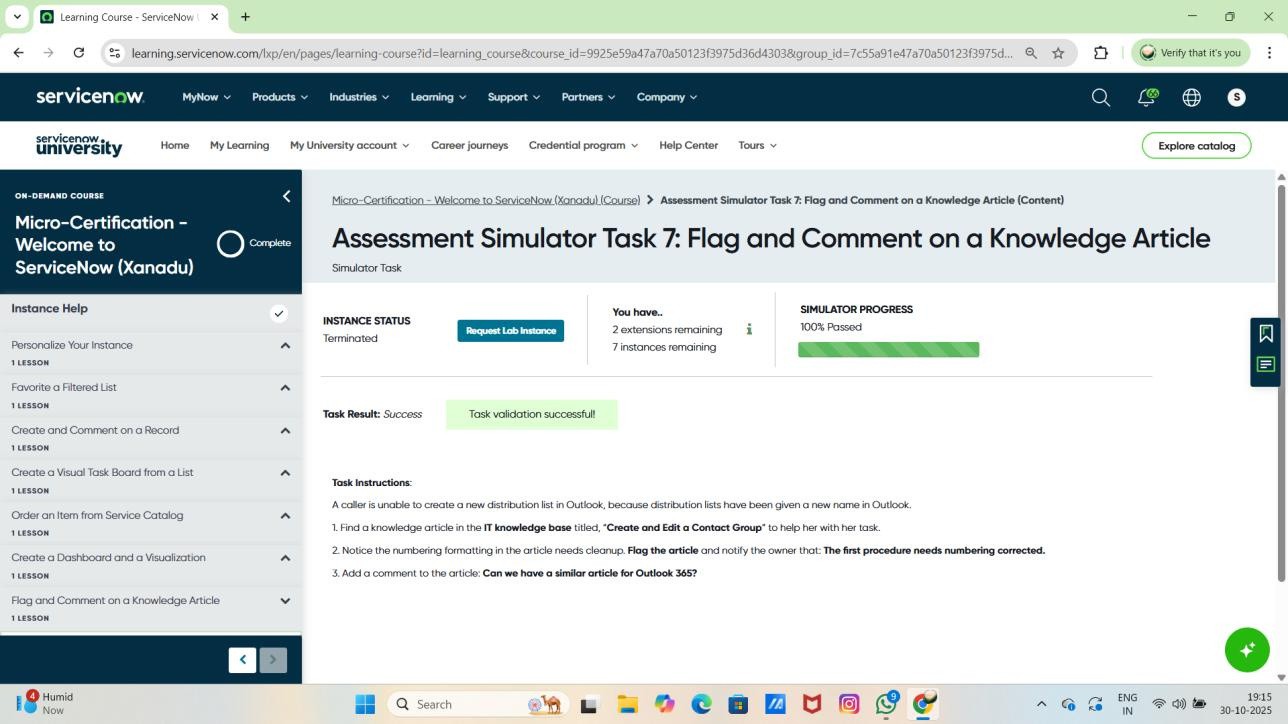
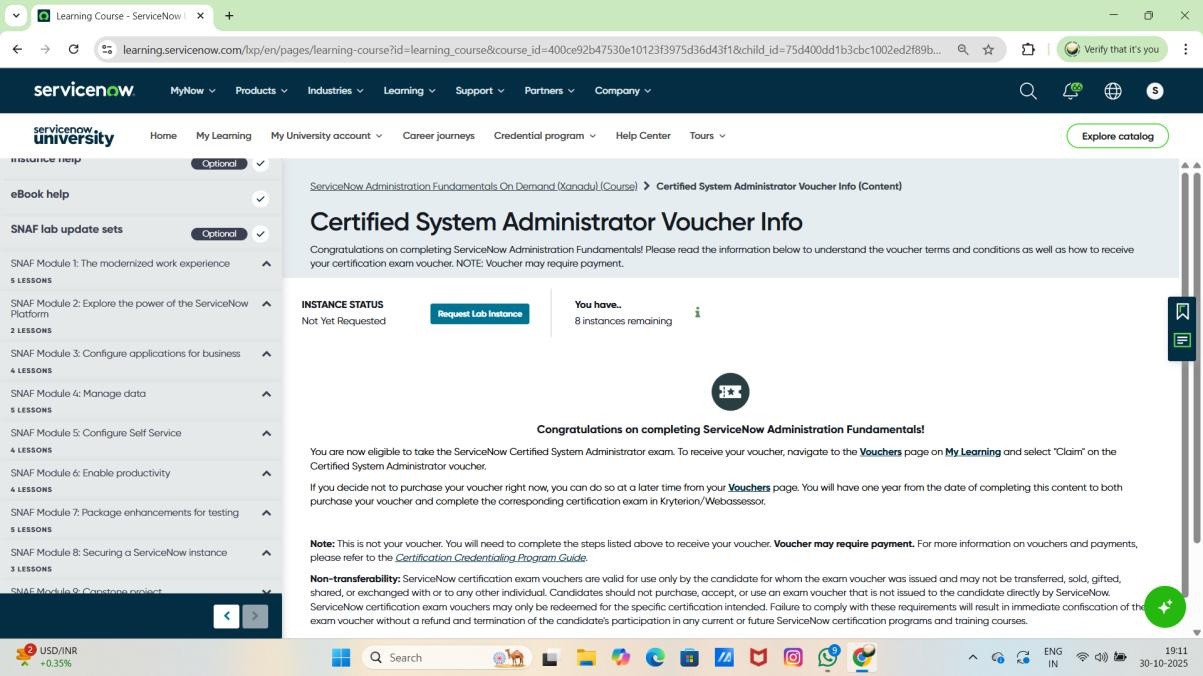
## ServiceNow Administrator:

**Understanding and Managing the ServiceNow Platform** – Learn how the ServiceNow platform works, its features, and how to explore its powerful tools for modern work management.

**Configuring Applications and Managing Data** – Gain skills to configure business applications, organize data, and manage system information efficiently.

**Improving Productivity and Self-Service** – Learn how to enable productivity features, create self- service portals, and automate workflows for better user experience.

**Securing and Testing the System** – Understand how to secure ServiceNow instances, package updates, and perform testing to ensure system reliability and performance.



## ServiceNow Micro-Certification:

**Working with Records and Lists** – Learn how to create, comment on, and manage records, as well as use filteredlistseffectively.

**Personalizing the ServiceNow Instance** – Understand how to customize your workspace, dashboards, and viewsbasedonyourneeds.

**Using Service Catalog and Task Boards** – Gain skills to order items from the Service Catalog and create visual task boards for better workflow management.

**Collaborating through Knowledge and Dashboards** – Learn to create dashboards, visualizations, and comment on knowledge articles to improve teamwork and communication.

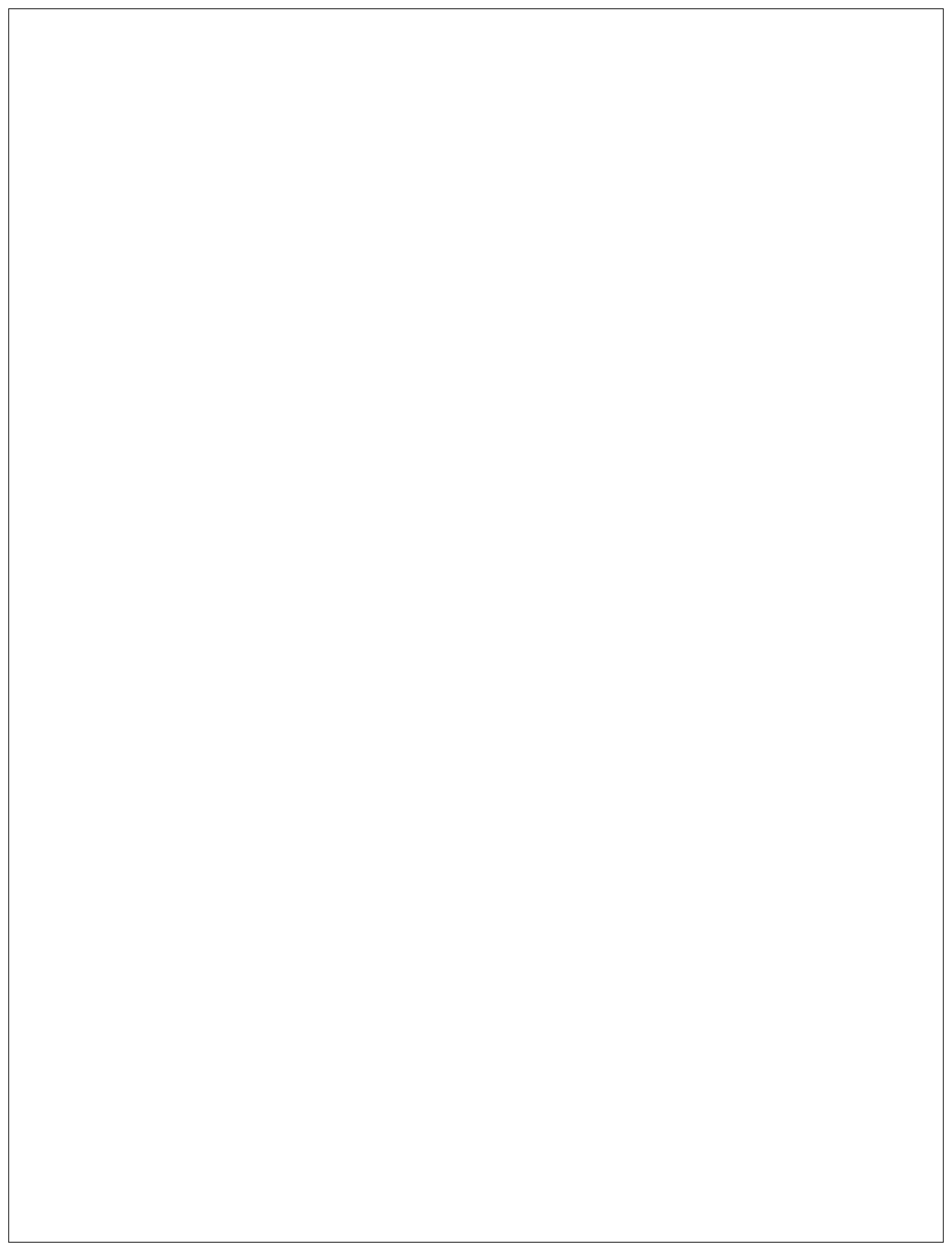
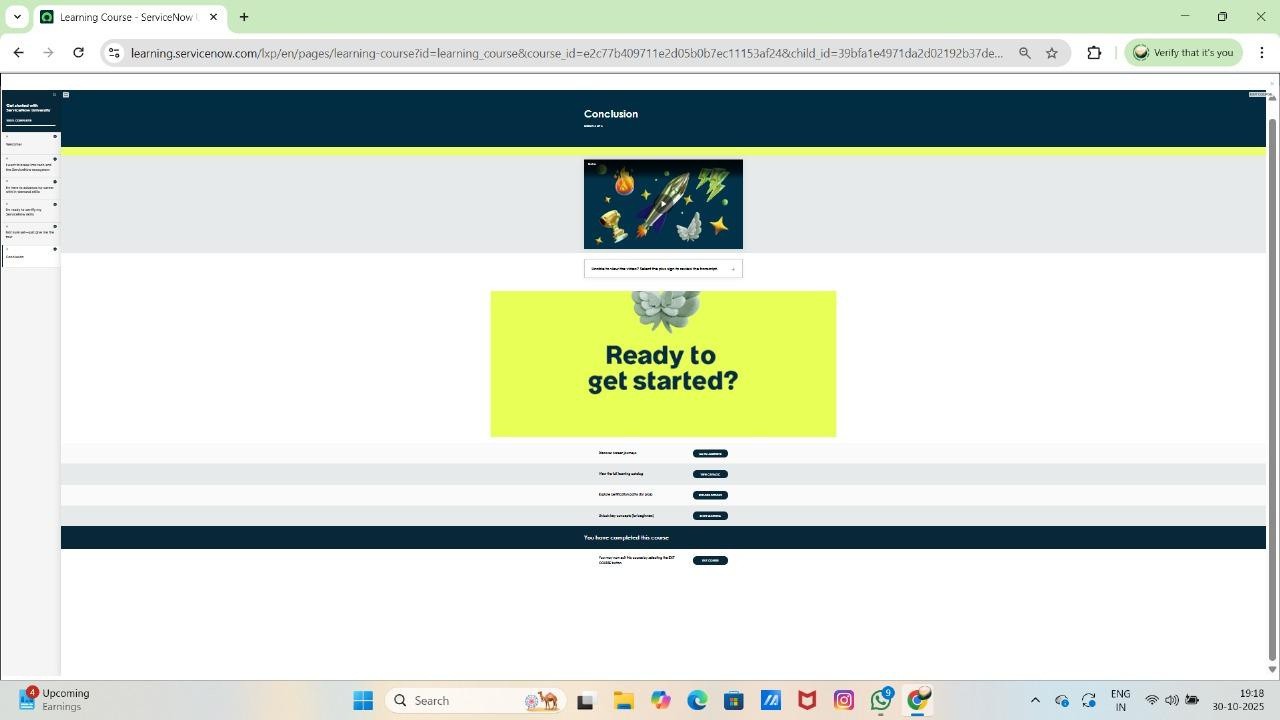
## Get Started with ServiceNow University:

### Introduction to ServiceNow University

supports your learning journey.

– Understand what ServiceNow University is and how it

**Navigating the Learning Platform** –Learn how to browse, access, and enroll in courses, certifications, and learningpaths.



**Tracking Progress and Achievements** – Discover how to monitor your course progress and earn badges or certifications.

**Using Learning Resources Effectively** – Learn to use available tools, eBooks, and support materials to enhanceyourServiceNow skills.

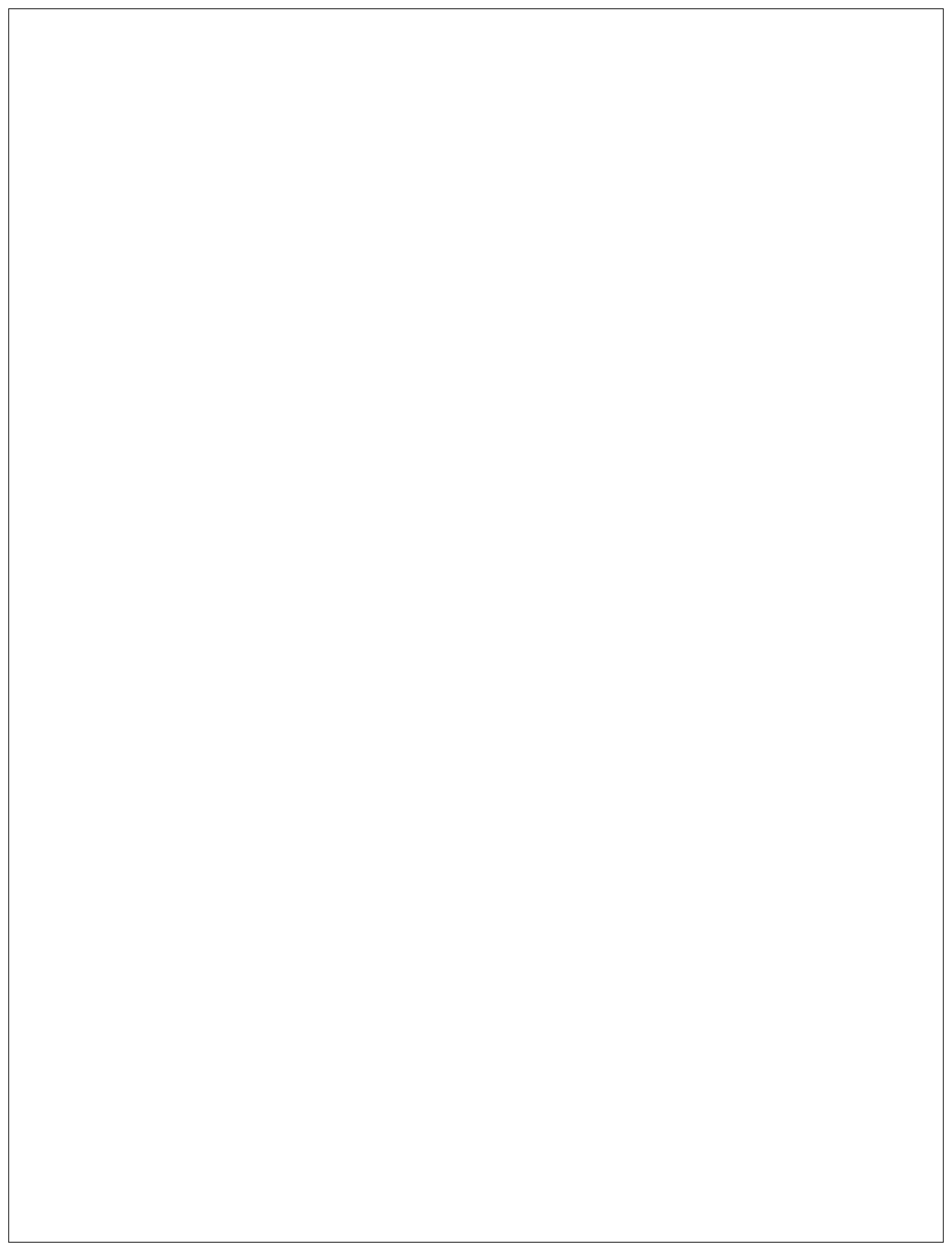
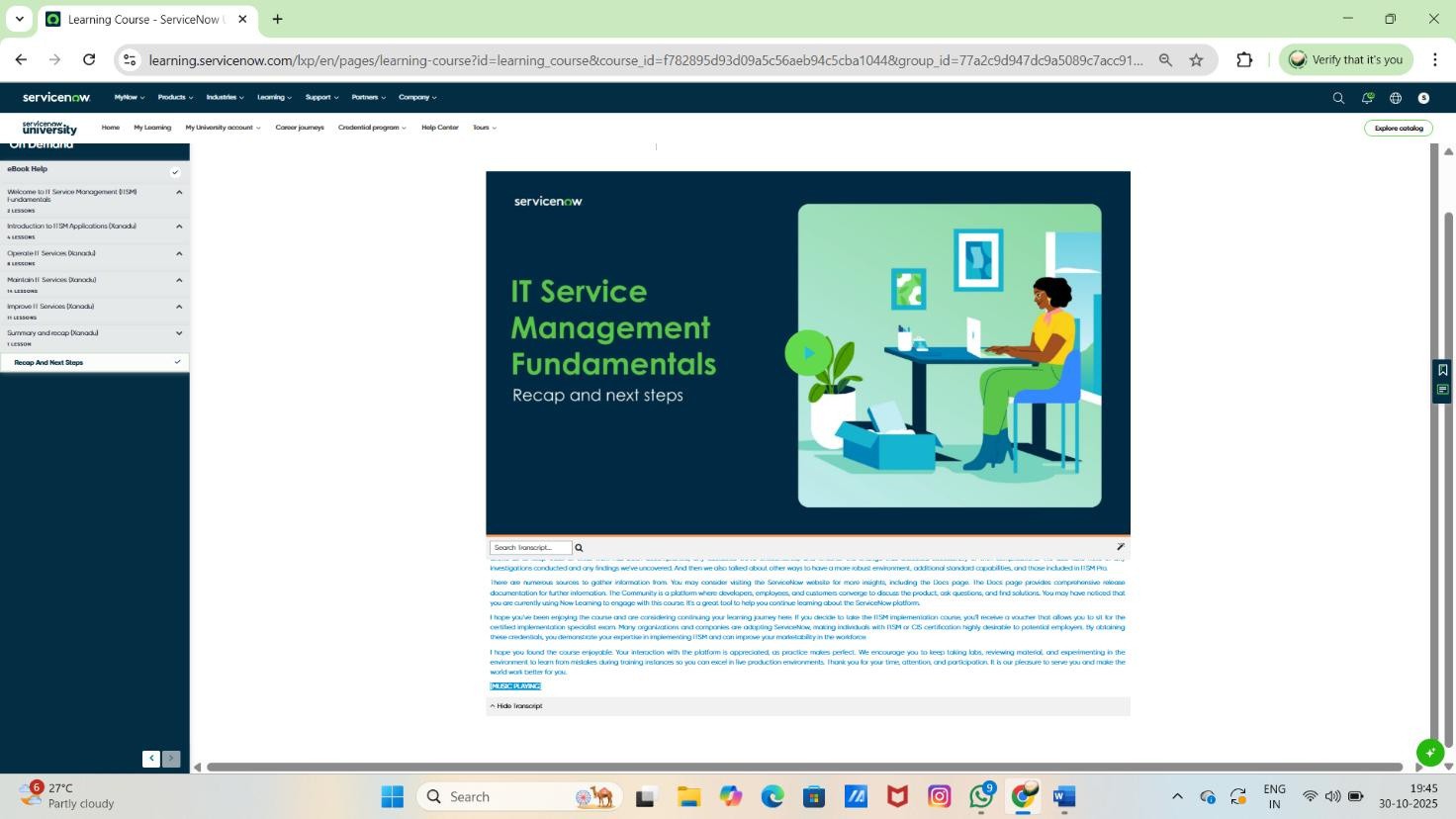
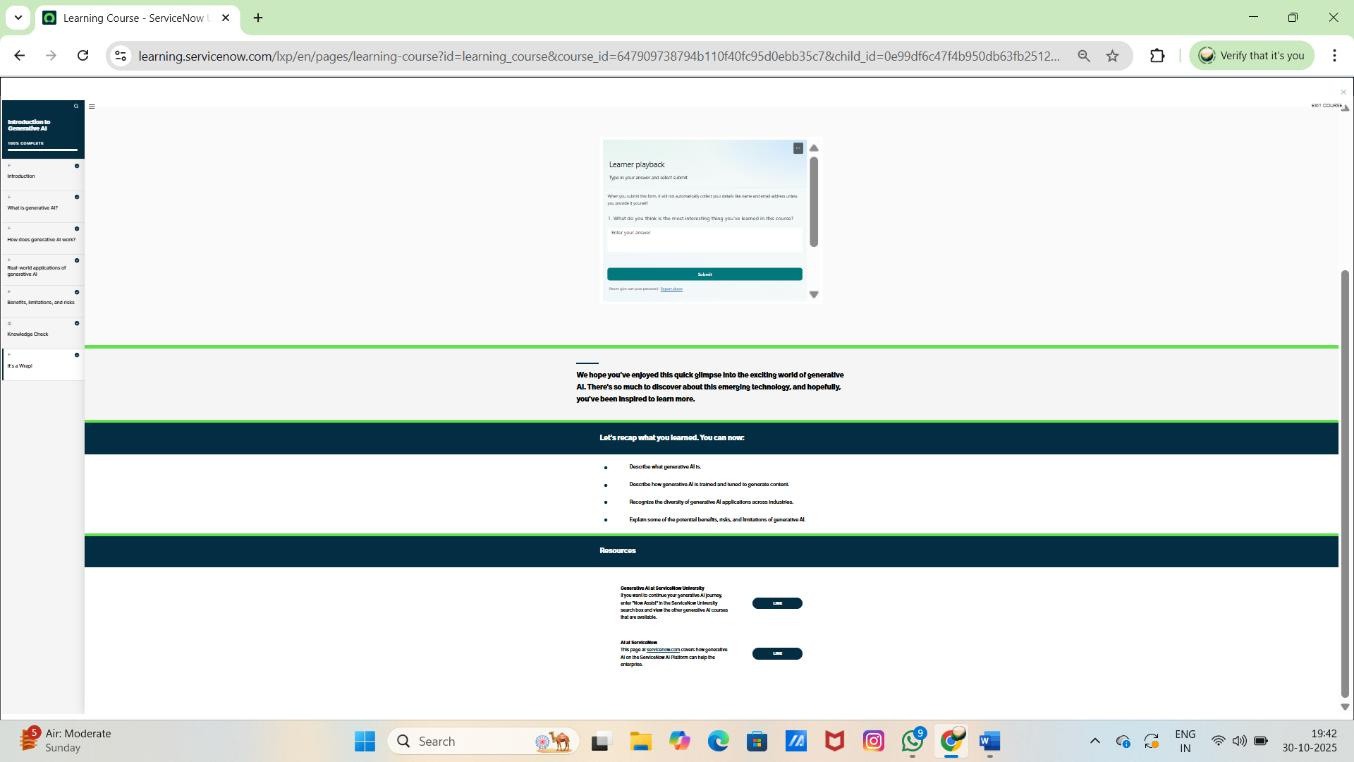
## Introduction to Generative AI:

**Basics of Generative AI** – Understand what Generative AI is and how it differs from traditional artificial intelligence.

**Working Principles** – Learn how Generative AI models create new content such as text, images, or code using machine learning techniques.

**Applications in Real Life** – Explore how Generative AI is used in areas like chatbots, content creation, automation, and business innovation.

**Ethical Use and Future Scope** – Gain awareness about responsible use, data ethics, and the future potentialof Generative AI in technology.



## IT Service Management (ITSM) Fundamentals:

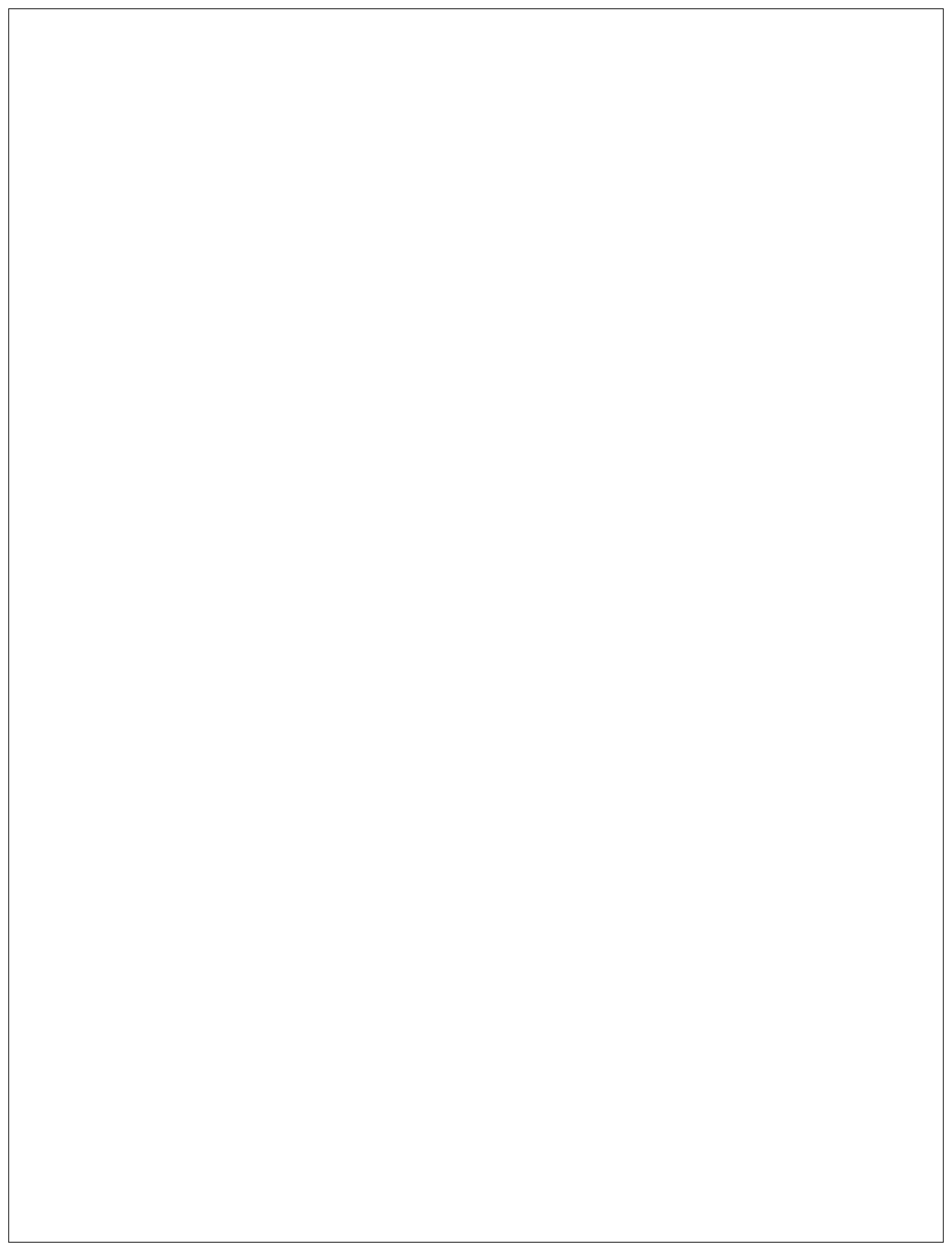
**Understanding ITSM Concepts** – Learn the basics of IT Service Management, including its goals, processes, and importance in delivering quality IT services.

**Operating and Maintaining IT Services** – Gain knowledge about how to run daily IT operations, monitor performance, and maintain service reliability.

**Improving IT Services** – Understand methods to identify service issues and apply continuous improvement strategies for better efficiency.

**Using ITSM Applications in ServiceNow** – Learn to use ServiceNow tools for managing incidents, requests, problems, and service performance effectively.

## UI Builder Fundamentals:



**Understanding UI Builder Basics** – Learn what UI Builder is and how it helps in designing and customizing user interfaces in ServiceNow.

**CreatingandConfiguring Pages** – Gain skills to create, edit, and manage pages using layouts, components, and data sources.

**Designing Interactive Experiences** – Understand how to build user-friendly and responsive interfaces for betterworkflowandnavigation.

**Using Data and Components Effectively** – Learn to connect data from different sources and use components to display and manage information dynamically.

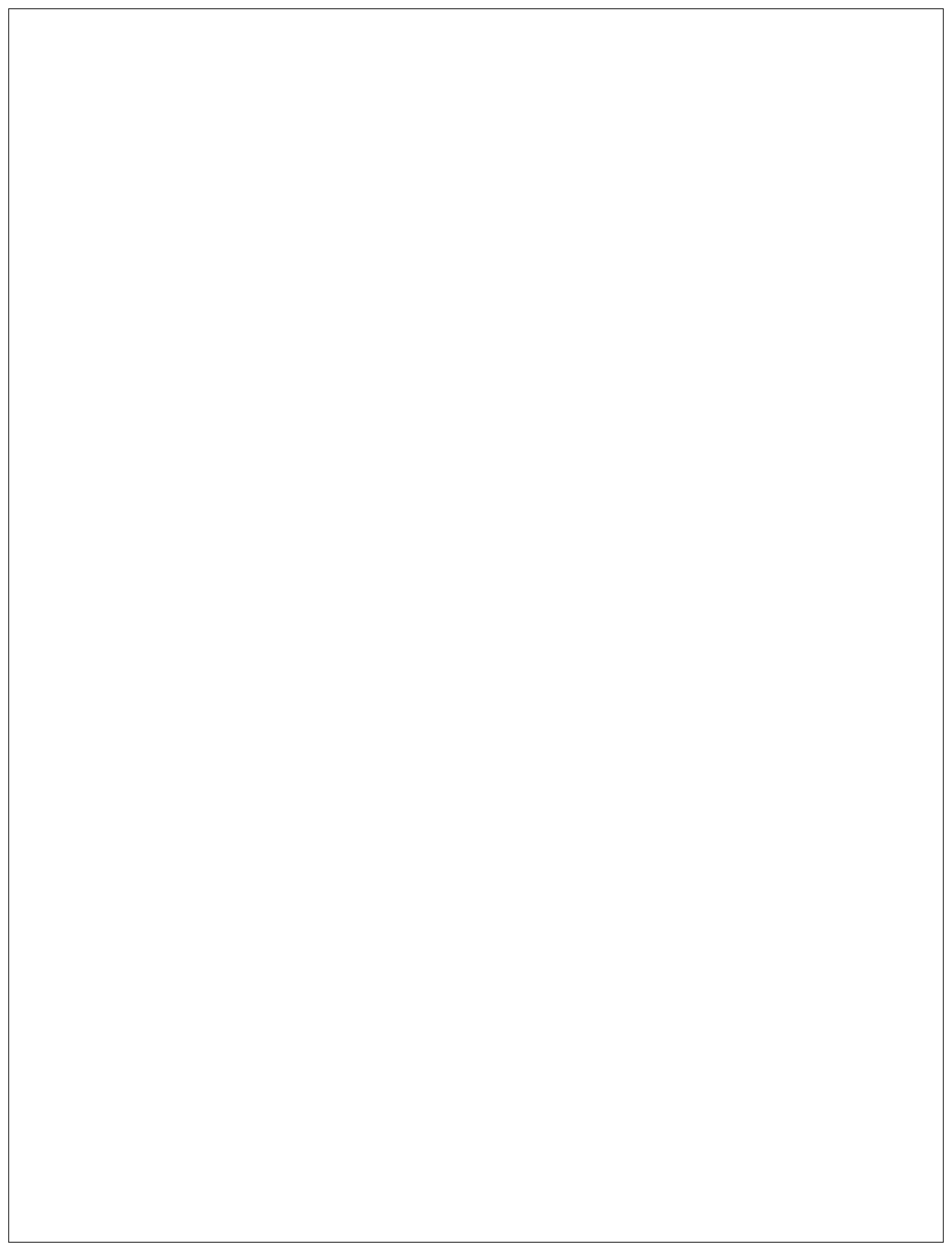
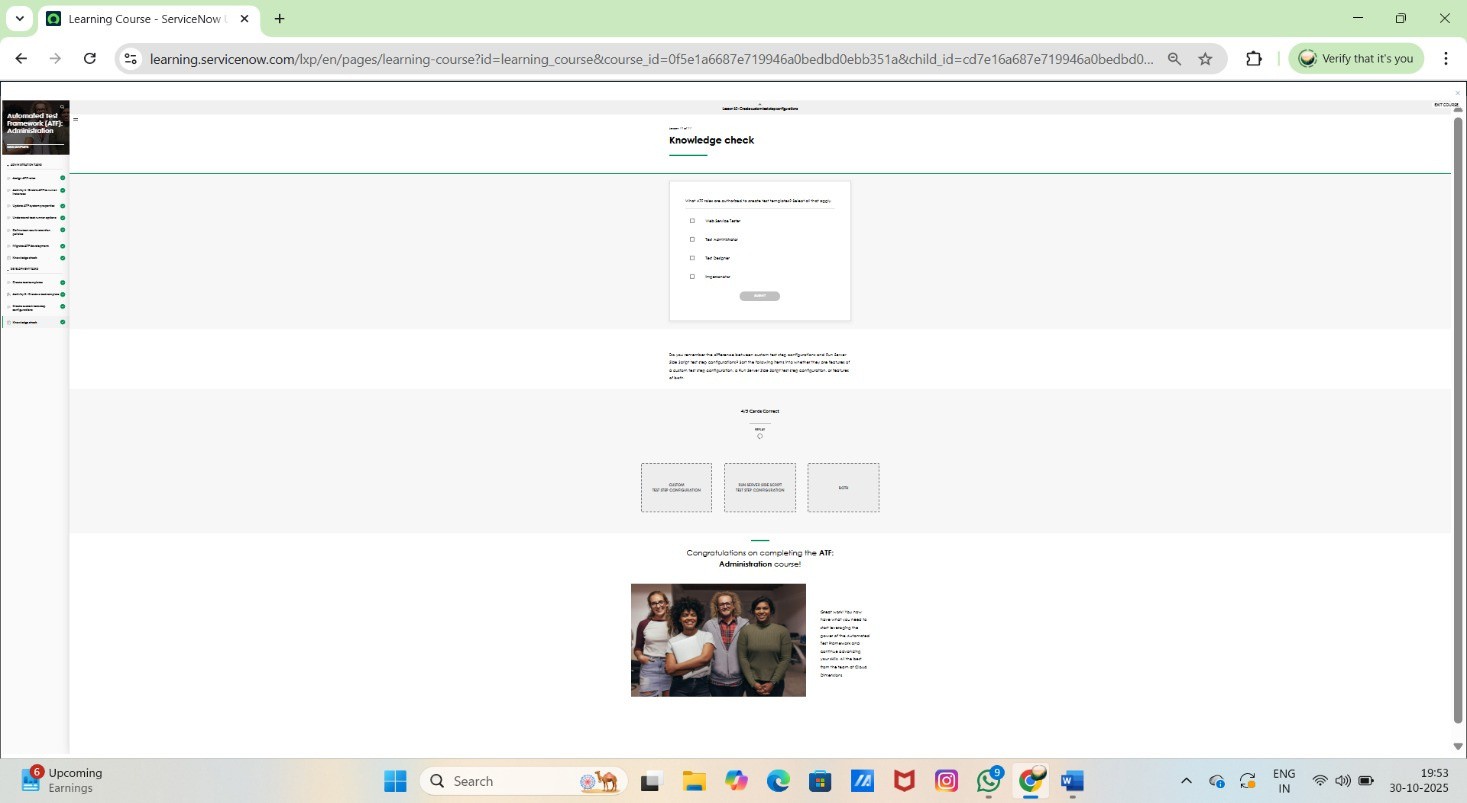
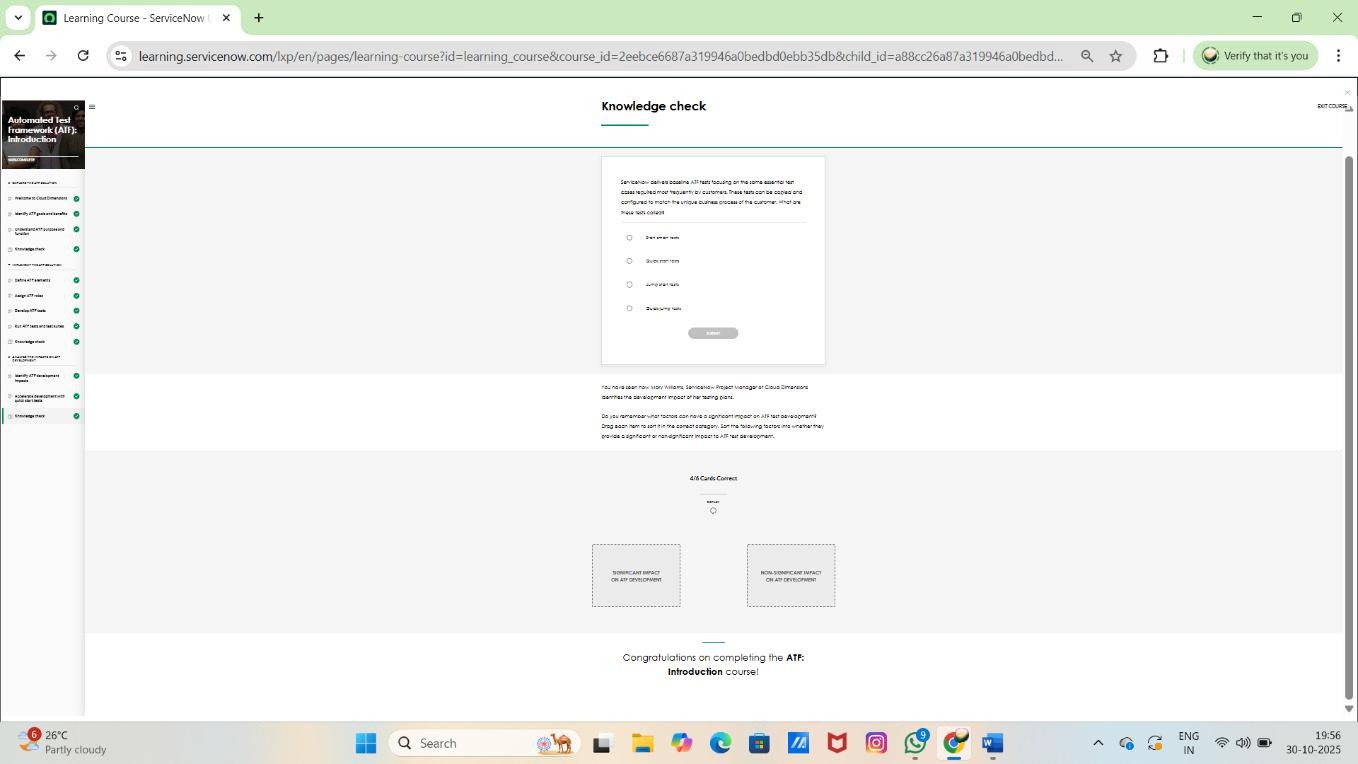
## Automated Test Framework (ATF): Introduction:

**Understanding ATF Basics** – Learn what the Automated Test Framework is and how it helps in testing ServiceNow applications.

**Creating and Running Tests** – Gain knowledge on how to create, configure, and execute automated tests tochecksystemfunctionality.

**Improving Testing Efficiency** – Understand how automation saves time by detecting issues early and reducing manual testing efforts.

**Ensuring System Quality** – Learn how ATF helps maintain application stability, accuracy, and quality during updatesanddevelopment.



## Automated Test Framework (ATF): Administration:

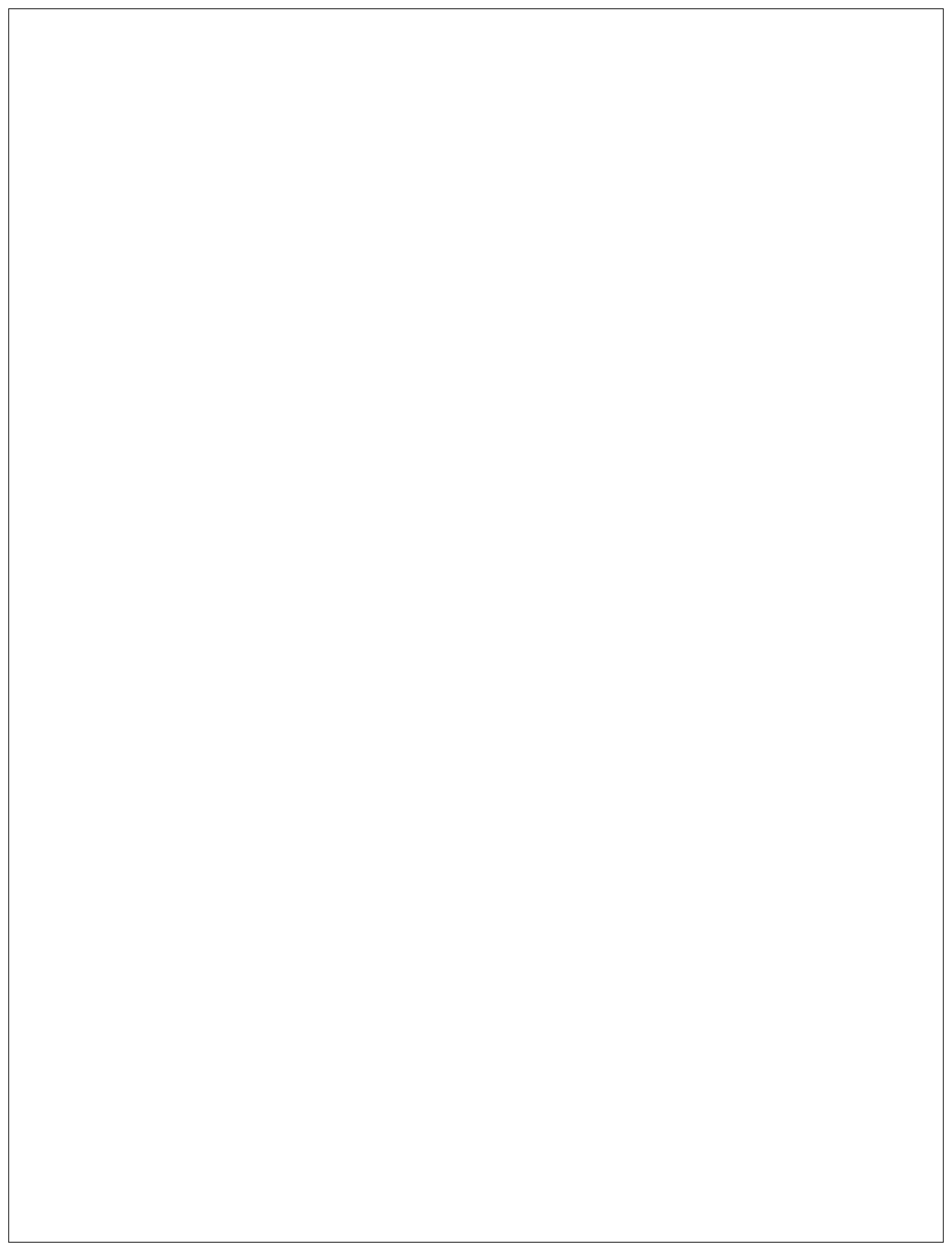
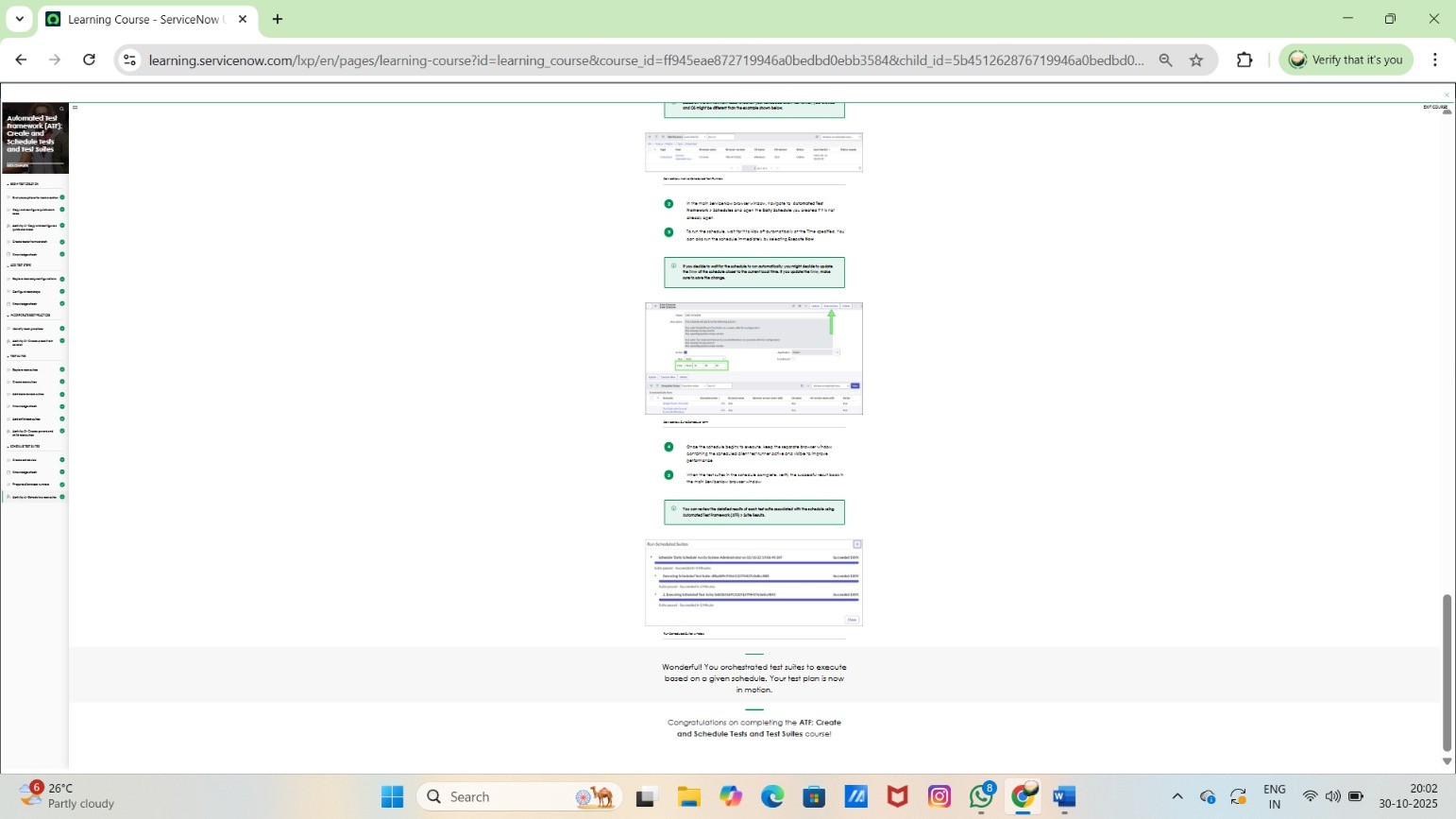
**How to Request and Use a Lab Instance** – Learn how to access and set up a ServiceNow lab environment for practicing ATF exercises safely.

**Understanding ATF Administration** – Gain knowledge about configuring, managing, and maintaining Automated Test Framework settings in ServiceNow.

**CreatingandManagingAutomated Tests** – Learn to design, organize, and execute test cases to ensure application quality and reliability.

**Monitoring and Troubleshooting Test Results** – Understand how to review test results, identify errors, and improvetestperformanceefficiently.

## Automated Test Framework (ATF): Create and Schedule Tests and Test Suites



**Requesting and Using a Lab Instance** – Learn how to access and set up a ServiceNow lab environment for hands-on practice with ATF.

**Creating Automated Tests** – Understand how to design and build automated test cases to verify ServiceNow application functionality.

**Buildingand Managing Test Suites** – Learn to group related tests into suites for efficient organization and bulk execution.

**Scheduling and Running Tests Automatically** – Gain skills to schedule test execution, monitor results, and ensureconsistentsystemperformance.