

Importing and Securing Data in ServiceNow

1. Introduction

Project Title: *Importing and Securing Data in ServiceNow*

Team ID: LTVIP2026TMIDS64983

Team Size: 4

- **Team Leader:** Doddamaneni Greeshma Sree– Project Lead & ServiceNow Developer
- **Team Member:** Gardimalla Anusha Reddy – Data Import , Transform Map Specialist,Business Rules
- **Team Member:** Perrala Akhila – Automation Developer,Security &ACL configuration Engineer
- **Team Member :**Putti Hyndavi – Documentation & Reporting Analyst

2.Project Overview

2.1. Purpose

The purpose of this project is to efficiently import external data into ServiceNow and securely associate each record with an employee. By linking records to employees and automatically fetching employee-related details such as department, the system improves data ownership, security, and reporting accuracy.

This project focuses on real-world enterprise challenges where large volumes of data are imported from external sources like Excel or CSV files. Without proper linkage and security, such data becomes difficult to manage and analyze.

2.2. Features

- Importing data using Import Sets
- Linking each record to an employee (sys_user)
- Auto-populating department details
- Implementing Access Control Lists (ACLs)
- Secure and role-based data access
- Department-wise and employee-wise reporting

3. Architecture

3.1. Frontend

The frontend of this project is the ServiceNow platform UI, which provides standard forms, lists, and dashboards. Users interact with the system through ServiceNow forms to view and manage records securely based on access permissions.

The UI supports filtering, sorting, and reporting, making it easy for users and administrators to analyze data without additional frontend development.

3.2. Backend

The backend is managed entirely within the ServiceNow platform using server-side components such as Transform Maps, Business Rules, and Access Control Rules. These components handle data processing, automation, and security.

Business logic is implemented using server-side JavaScript to ensure data integrity and automation during record creation and updates.

3.3 Database

ServiceNow uses its internal relational database. Custom tables are created to store imported records, while reference fields link data to existing tables such as `sys_user` and `cmn_department`.

4. Setup Instructions

4.1. Prerequisites

- ServiceNow Personal Developer Instance
- Basic knowledge of ServiceNow navigation
- CSV or Excel file containing employee-related data

4.2. Installation

- Log in to the ServiceNow Developer Instance
- Create a custom table for storing imported data
- Upload external data using Import Sets
- Configure Transform Maps for field mapping
- Test data import and validation

5. Folder Structure

Application Scope

- Custom Tables
- Import Sets
- Transform Maps
- Business Rules
- Access Control Rules (ACLs)
- Reports and Dashboards

This modular structure ensures maintainability and scalability of the application.

6. Running the Application

Once configured, the application runs automatically within the ServiceNow instance. Users can:

- Import data using Import Sets
- View records via list and form views
- Access reports and dashboards

No external server startup is required.

7. API Documentation

In this project, APIs are used to enable interaction between external data sources and the ServiceNow platform. ServiceNow provides **Table APIs** and **Import Set APIs** that allow secure creation, retrieval, updating, and deletion of records in tables. These APIs help in automating data import and ensuring seamless integration with external systems such as HR databases or third-party applications.

The API layer plays a crucial role in ensuring that data is transferred securely and consistently into the system. Authentication and authorization mechanisms ensure that only valid users or systems can access the APIs. This makes the solution scalable and suitable for enterprise-level data handling.

7.1 Import Set API

The **Import Set API** is used to load external data (CSV/Excel or external system data) into a staging table in **ServiceNow**. Once the data is loaded, a Transform Map processes and inserts the data into the target custom table.

Purpose:

- Import large volumes of external data
- Stage data before validation and transformation

Method:
POST

Example Use Case:

Uploading employee-related data such as employee email, ID, or other attributes from an external file.

7.2 Table API

The **Table API** allows interaction with ServiceNow tables using RESTful web services. It supports standard CRUD operations on both system tables and custom tables created for this project.

Operations Supported:

- Create new records
- Retrieve existing records
- Update records
- Delete records (based on access permissions)

Purpose:

- Fetch employee-linked records
- Update department or ownership information
- Enable reporting and auditing

7.3 API Authentication

API authentication is handled using ServiceNow's built-in authentication mechanisms. Only authenticated users or systems with proper roles can access the APIs.

Authentication Methods Used:

- Basic Authentication
- Role-based access control

This ensures that sensitive employee data is protected and unauthorized access is prevented.

7.4 Security in APIs

Security is enforced at the API level using:

- Access Control Lists (ACLs)

- Role validation
- User context checks

Even if an API call is successful, data visibility depends on ACL conditions. For example, employees can access only their own records, while administrators have full access.

7.5 API Benefits in the Project

- Enables seamless data import from external sources
- Supports automation and integration
- Ensures secure data handling
- Improves scalability of the solution
- Facilitates future integrations with HR or ERP systems

8. Authentication

Authentication is handled by ServiceNow's built-in user authentication mechanism. Authorization is enforced using Access Control Lists (ACLs).

ACLs ensure that:

- Users can view only their own records
- Admins have full access
- Unauthorized access is restricted

This guarantees data confidentiality and integrity.

9. User Interface

The user interface includes:

- Record list views
- Detailed form views
- Filtered reports
- Department-wise dashboards

The interface is intuitive and user-friendly, requiring minimal training.

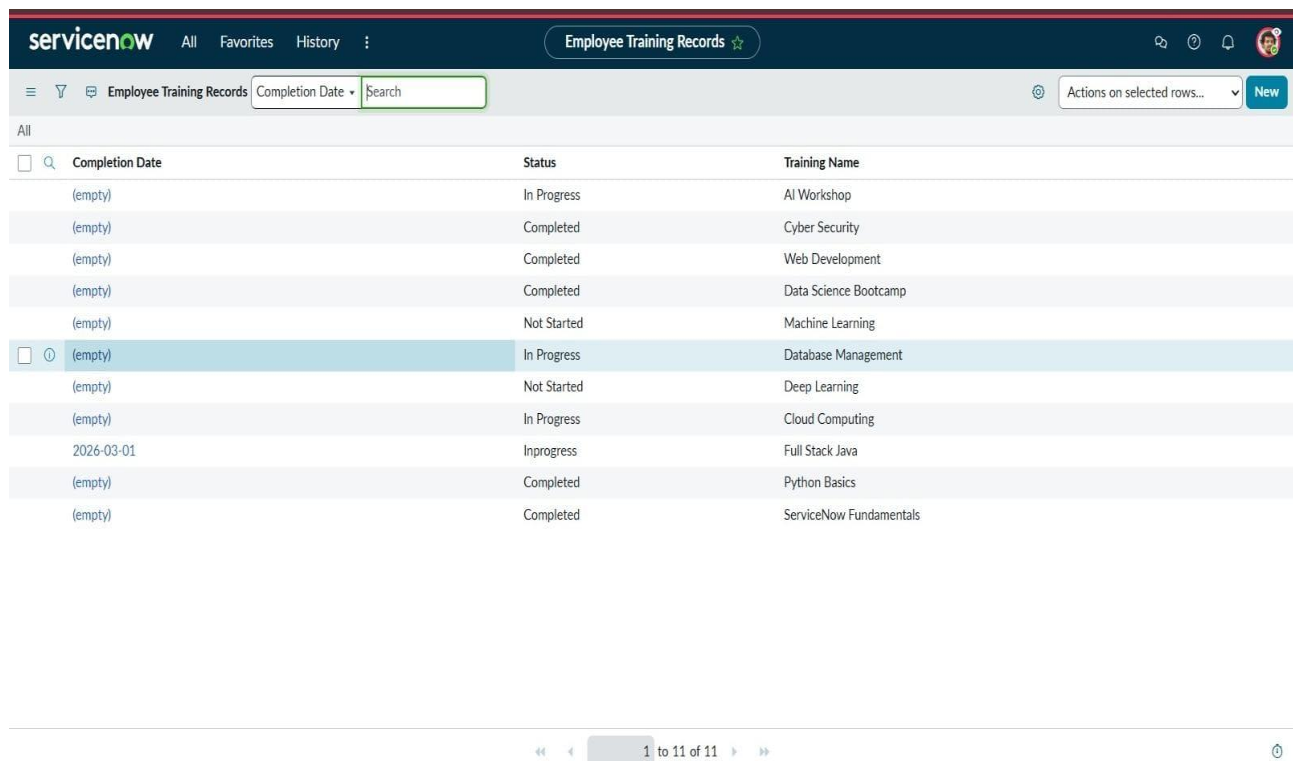
10. Testing

Testing was performed using:

- Data import validation
- User access testing
- Role-based security testing
- Report accuracy verification

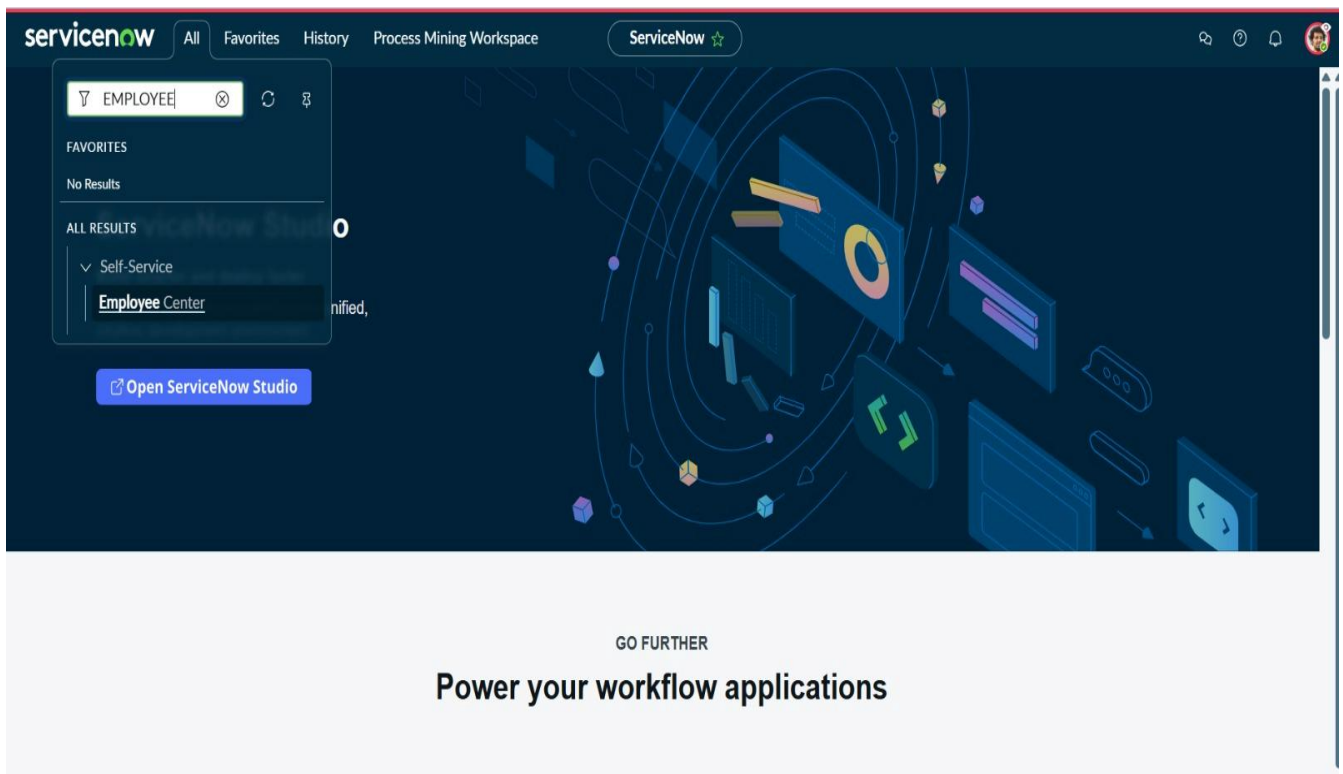
Each module was tested to ensure correct functionality and data security.

11. Screenshots



The screenshot shows the ServiceNow interface for the 'Employee Training Records' table. The table has three columns: 'Completion Date', 'Status', and 'Training Name'. There are 11 rows of data. The row for 'Database Management' is selected. The table is displayed in a list view with a search bar and a 'New' button.

Completion Date	Status	Training Name
(empty)	In Progress	AI Workshop
(empty)	Completed	Cyber Security
(empty)	Completed	Web Development
(empty)	Completed	Data Science Bootcamp
(empty)	Not Started	Machine Learning
(empty)	In Progress	Database Management
(empty)	Not Started	Deep Learning
(empty)	In Progress	Cloud Computing
2026-03-01	Inprogress	Full Stack Java
(empty)	Completed	Python Basics
(empty)	Completed	ServiceNow Fundamentals



12. Known Issues

- Incorrect employee data may cause failed mappings
- Missing reference fields can affect reports
- Data format issues in CSV files

These issues can be resolved through proper validation and error handling.

13. Future Enhancements

- Integration with external HR systems
- Automated data imports using scheduled jobs
- Advanced analytics dashboards
- Email notifications for record updates

These enhancements will further improve automation and usability.