Phase 8: Data Management & Deployment

8.1Data Import Wizard & Data Loader

Two primary tools for data operations were evaluated:

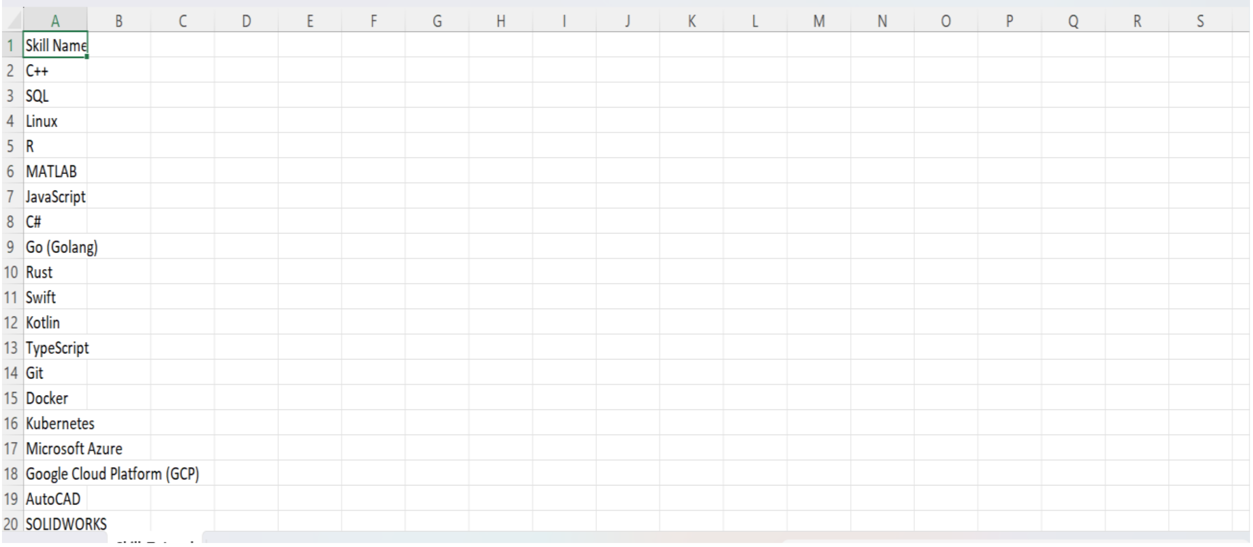
Data Import Wizard: An in-browser, user-friendly tool for simple data loads of less than 50,000 records. It supports standard objects and custom objects.

Data Loader: A more powerful, client-side application for bulk data operations (Insert, Update, Upsert, Delete) of up to 5 million records. It provides more control over mapping and error handling.

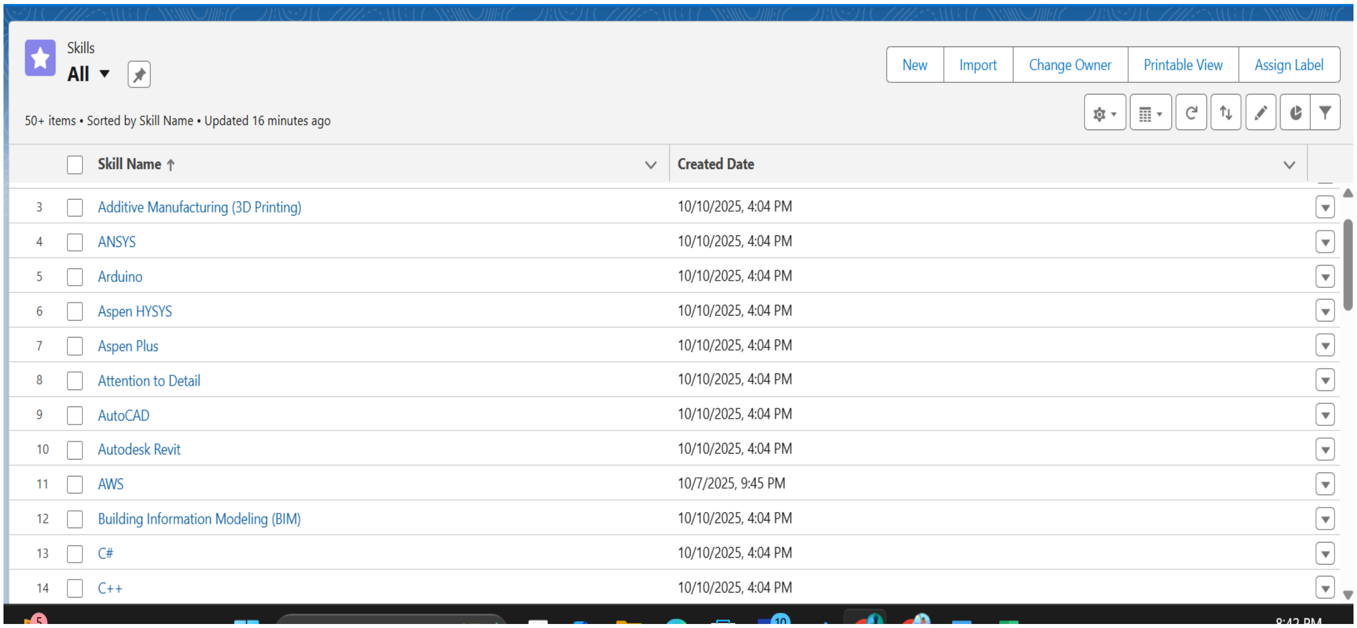
Chosen Tool: Data Loader was selected for all data management tasks in this project due to its robustness and the need to perform multiple types of operations (Insert and Delete).

Implementation - Bulk Deletion: Data Loader was initially used to perform a bulk delete operation. An Export was run on the Skill__c object with a WHERE clause (Name LIKE 'a0%') to get the IDs of incorrectly created records. These IDs were then used in a Delete operation to clean the database.

- The query:
  - **Field:** Name
  - **Operator:** starts with
  - **Value:** a0
- Then imported the skills by mapping using **Auto-Match Fields to Columns.**



	A
1	Skill Name
2	C++
3	SQL
4	Linux
5	R
6	MATLAB
7	JavaScript
8	C#
9	Go (Golang)
10	Rust
11	Swift
12	Kotlin
13	TypeScript
14	Git
15	Docker
16	Kubernetes
17	Microsoft Azure
18	Google Cloud Platform (GCP)
19	AutoCAD
20	SOLIDWORKS



Skills

All ▾

50+ items • Sorted by Skill Name • Updated 16 minutes ago

	□	Skill Name ↑	Created Date
3	□	Additive Manufacturing (3D Printing)	10/10/2025, 4:04 PM
4	□	ANSYS	10/10/2025, 4:04 PM
5	□	Arduino	10/10/2025, 4:04 PM
6	□	Aspen HYSYS	10/10/2025, 4:04 PM
7	□	Aspen Plus	10/10/2025, 4:04 PM
8	□	Attention to Detail	10/10/2025, 4:04 PM
9	□	AutoCAD	10/10/2025, 4:04 PM
10	□	Autodesk Revit	10/10/2025, 4:04 PM
11	□	AWS	10/7/2025, 9:45 PM
12	□	Building Information Modeling (BIM)	10/10/2025, 4:04 PM
13	□	C#	10/10/2025, 4:04 PM
14	□	C++	10/10/2025, 4:04 PM

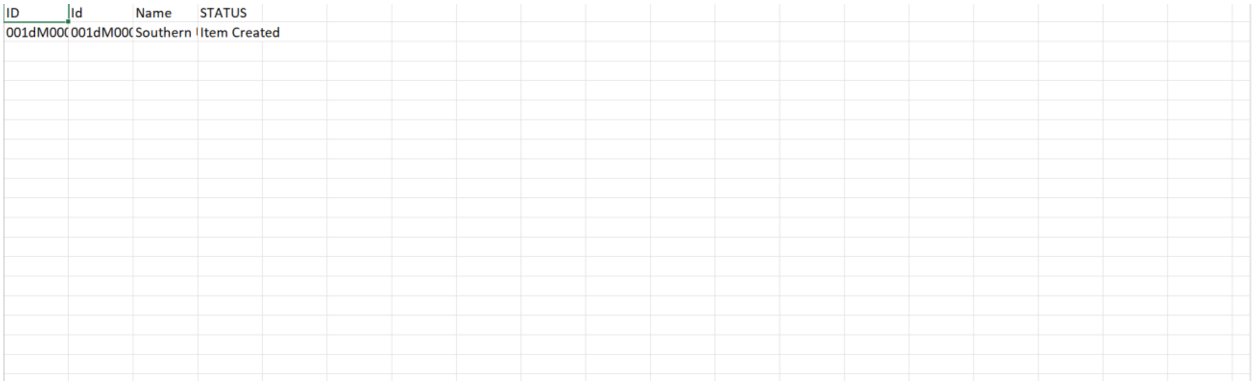
Implementation - Data Migration:

8.2 Data Export & Backup

Implementation: The **Data Export** feature in Data Loader was used for our data migration tasks. For a full-org backup, the best practice is to use the native **"Data Export Service"** found in Setup. This service allows an administrator to schedule a weekly or monthly export of all the organization's data into a set of CSV files. For this project, a manual export of key setup data was sufficient.

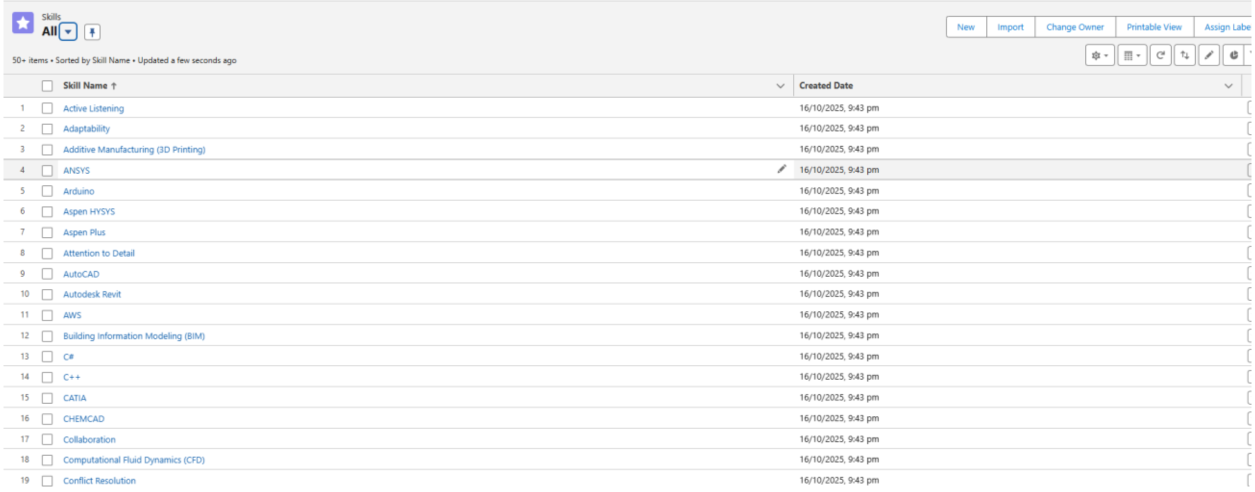
Data Loader was used for the critical post-deployment task of migrating setup data.

Accounts: The primary "Southern University" Account record was exported from the Source Org and then inserted into the Target Org. This was a crucial step, as this record is the parent for all Student records.



ID	Id	Name	STATUS
001dM00(	001dM00(	Southern I	Item Created

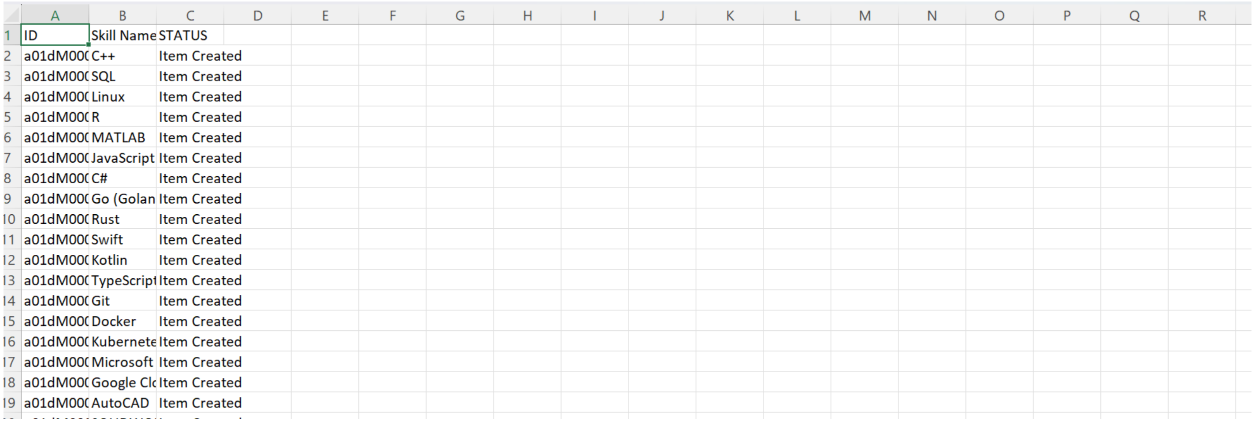
Skills: The master list of all Skill__c records was exported from the Source Org (querying only the Name field) and then inserted into the Target Org to populate the skill library.



Skills
All

50+ items • Sorted by Skill Name • Updated a few seconds ago

		Skill Name ↑	Created Date
1	□	Active Listening	16/10/2025, 9:43 pm
2	□	Adaptability	16/10/2025, 9:43 pm
3	□	Additive Manufacturing (3D Printing)	16/10/2025, 9:43 pm
4	□	ANSYS	16/10/2025, 9:43 pm
5	□	Arduino	16/10/2025, 9:43 pm
6	□	Aspen HYSYS	16/10/2025, 9:43 pm
7	□	Aspen Plus	16/10/2025, 9:43 pm
8	□	Attention to Detail	16/10/2025, 9:43 pm
9	□	AutoCAD	16/10/2025, 9:43 pm
10	□	Autodesk Revit	16/10/2025, 9:43 pm
11	□	AWS	16/10/2025, 9:43 pm
12	□	Building Information Modeling (BIM)	16/10/2025, 9:43 pm
13	□	C#	16/10/2025, 9:43 pm
14	□	C++	16/10/2025, 9:43 pm
15	□	CATIA	16/10/2025, 9:43 pm
16	□	CHEMCAD	16/10/2025, 9:43 pm
17	□	Collaboration	16/10/2025, 9:43 pm
18	□	Computational Fluid Dynamics (CFD)	16/10/2025, 9:43 pm
19	□	Conflict Resolution	16/10/2025, 9:43 pm



ID	Skill Name	STATUS
a01dM000	C++	Item Created
a01dM000	SQL	Item Created
a01dM000	Linux	Item Created
a01dM000	R	Item Created
a01dM000	MATLAB	Item Created
a01dM000	JavaScript	Item Created
a01dM000	C#	Item Created
a01dM000	Go (Golan	Item Created
a01dM000	Rust	Item Created
a01dM000	Swift	Item Created
a01dM000	Kotlin	Item Created
a01dM000	TypeScript	Item Created
a01dM000	Git	Item Created
a01dM000	Docker	Item Created
a01dM000	Kubernete	Item Created
a01dM000	Microsoft	Item Created
a01dM000	Google Clc	Item Created
a01dM000	AutoCAD	Item Created

Change Sets vs. VS Code & SFDX

Two deployment methodologies were evaluated for moving the application's metadata.

Change Sets (Considered):

Concept: The standard, UI-based tool for moving metadata between connected orgs (typically a Sandbox and its Production org).

Analysis: We determined that Change Sets were not a viable option for this project. **Outbound Change Sets can only be sent from a Sandbox.** Since our development environment was a Developer Edition org, it could not create or send Outbound Change Sets.

VS Code & SFDX (Implemented):

This was the chosen professional-grade deployment method.

Visual Studio Code (VS Code): Used as the local Integrated Development Environment (IDE).

Salesforce Extension Pack: Installed in VS Code to provide SFDX commands and the Org Browser.

Salesforce CLI: The underlying engine used to communicate with the Salesforce orgs.

The SFDX Deployment Process (Our Final, Successful Strategy)

After significant troubleshooting, a robust, multi-stage deployment strategy using SFDX was executed.

Project Setup: A new, clean SFDX project was created locally.

Authentication: The Source (Dev) Org and the new Target (Dev) Org were both authorized using the SFDX: Authorize an Org command, creating local aliases (WeForYouCareSource, WeForYou CareTarget).

Manifest Creation (package.xml): A comprehensive package.xml manifest file was created. This file acted as the "shopping list," explicitly listing every component type and member (Apex Classes, Custom Objects, Flows, LWC, etc.) that made up the StudentCare application. This was a critical step to ensure a complete deployment package.

Metadata Retrieval: The SFDX: Retrieve Source in Manifest from Org command was run against the Source Org. This downloaded a complete, local copy of the application's metadata into the force-app directory.

Staged Deployment: Due to complex dependencies (specifically the CustomApplication referencing Profiles and other metadata), a single deployment was prone to "invalid cross reference id" errors. A two-stage deployment was implemented:

Stage 1 (package-stage1-core.xml): Deployed all foundational components (Objects, Apex, Flows, LWC) *except* for the CustomApplication, Profiles, and other security/UI-binding metadata.

Stage 2 (package-stage2-ui-and-security.xml): After Stage 1 succeeded, this package deployed the "finishing touches," including the CustomApplication and the page layouts.

Post-Deployment Configuration: After a successful metadata deployment, it was noted that **Profiles and Permission Sets were not deployed**. Critical post-deployment steps were performed manually in the Target Org:

Created a Permission Set(WeForYouAdmin Access) to grant the System Administrator access to all new custom objects, fields, and Apex classes.

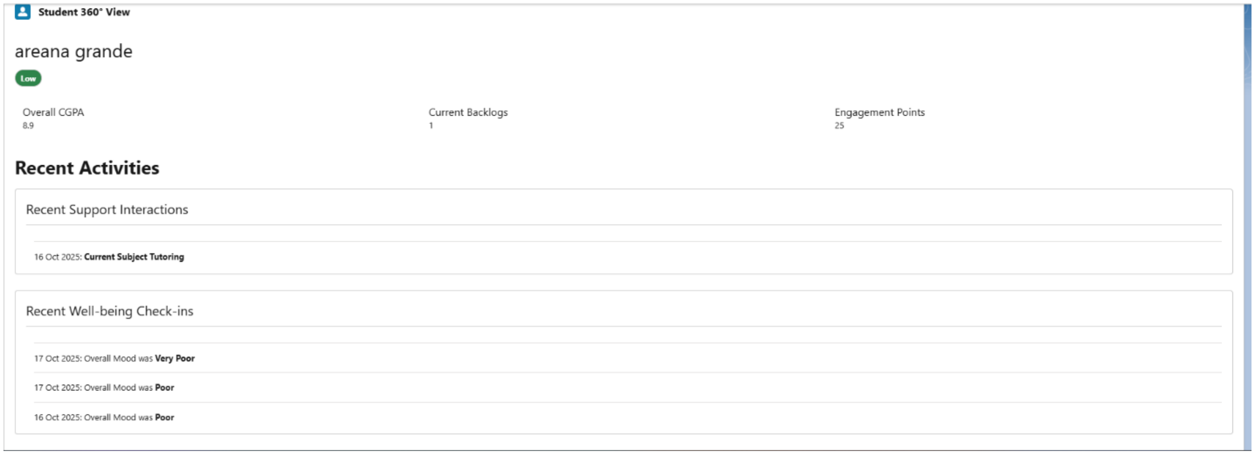
Manually assigned the WeForYou application to the System Administrator profile.

Manually rectified Profile assignments for Record Types and Tab Visibility that were missed during deployment. This was the final step that made the application fully usable for end-users.

ANT Migration Tool: This is an older, command-line tool for deploying metadata.
The **Salesforce CLI (used by SFDX) is the modern successor to the ANT Migration Tool** and is the recommended tool for all new development and deployment projects. We used the modern tool.

TARGET ORG CHECK:

1,LWC



Student 360° View

areana grande

Low

Overall CGPA	Current Backlogs	Engagement Points
8.9	1	25

Recent Activities

Recent Support Interactions

16 Oct 2025: **Current Subject Tutoring**

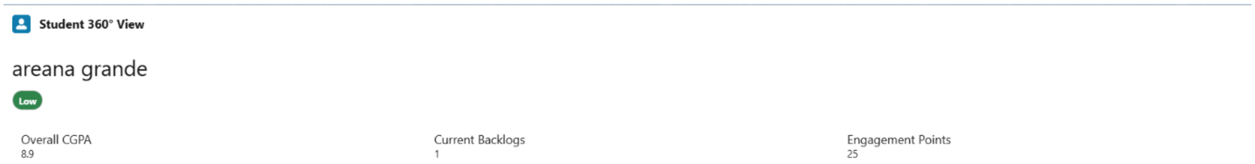
Recent Well-being Check-ins

17 Oct 2025: Overall Mood was **Very Poor**

17 Oct 2025: Overall Mood was **Poor**

16 Oct 2025: Overall Mood was **Poor**

2.Engagement points



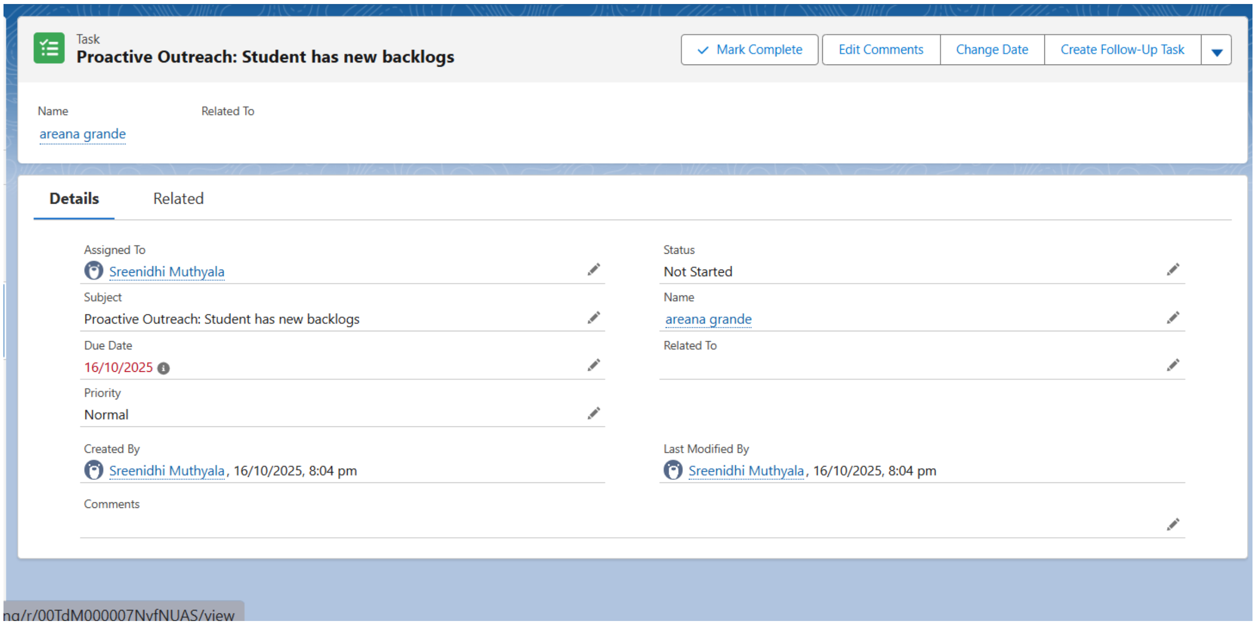
Student 360° View

areana grande

Low

Overall CGPA	Current Backlogs	Engagement Points
8.9	1	25

3.TASK Created for the mentor



Task

Proactive Outreach: Student has new backlogs

Name
areana grande

Related To

Assigned To
Sreenidhi Muthyala

Subject
Proactive Outreach: Student has new backlogs

Due Date
16/10/2025

Priority
Normal

Status
Not Started

Name
areana grande

Related To

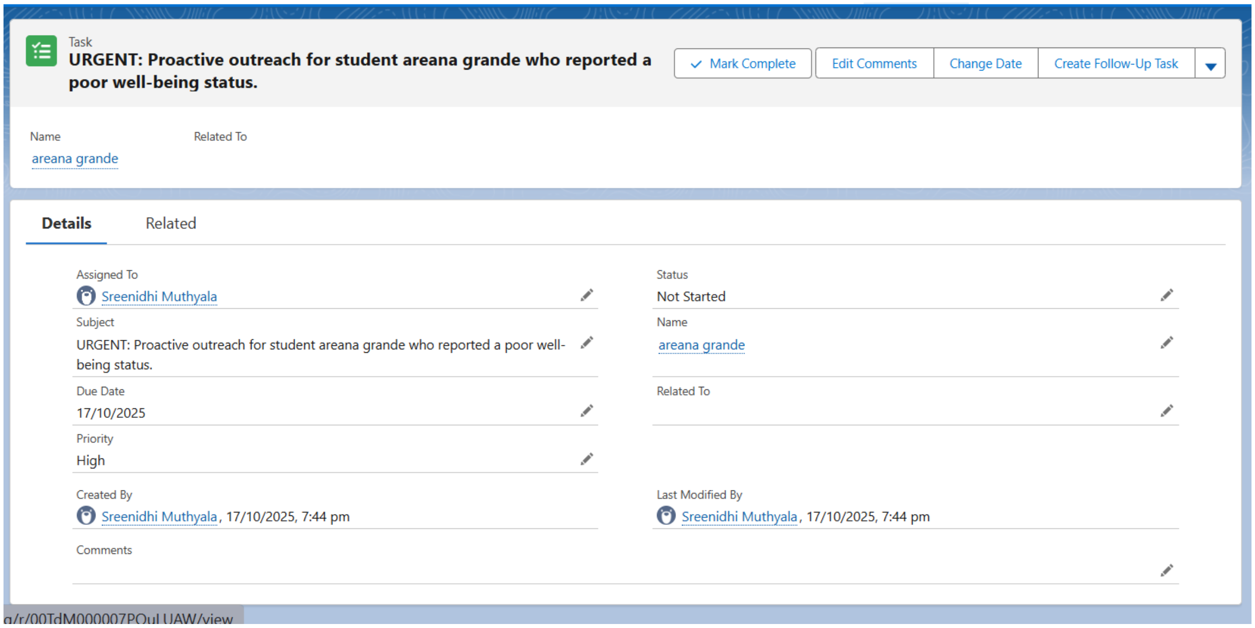
Created By
Sreenidhi Muthyala, 16/10/2025, 8:04 pm

Last Modified By
Sreenidhi Muthyala, 16/10/2025, 8:04 pm

Comments

ng/r/00TdM000007NvfNUAS/view

4.TASK Create to counselor



Task

URGENT: Proactive outreach for student areana grande who reported a poor well-being status.

Name
areana grande

Related To

Assigned To
Sreenidhi Muthyala

Subject
URGENT: Proactive outreach for student areana grande who reported a poor well-being status.

Due Date
17/10/2025

Priority
High

Status
Not Started

Name
areana grande

Related To

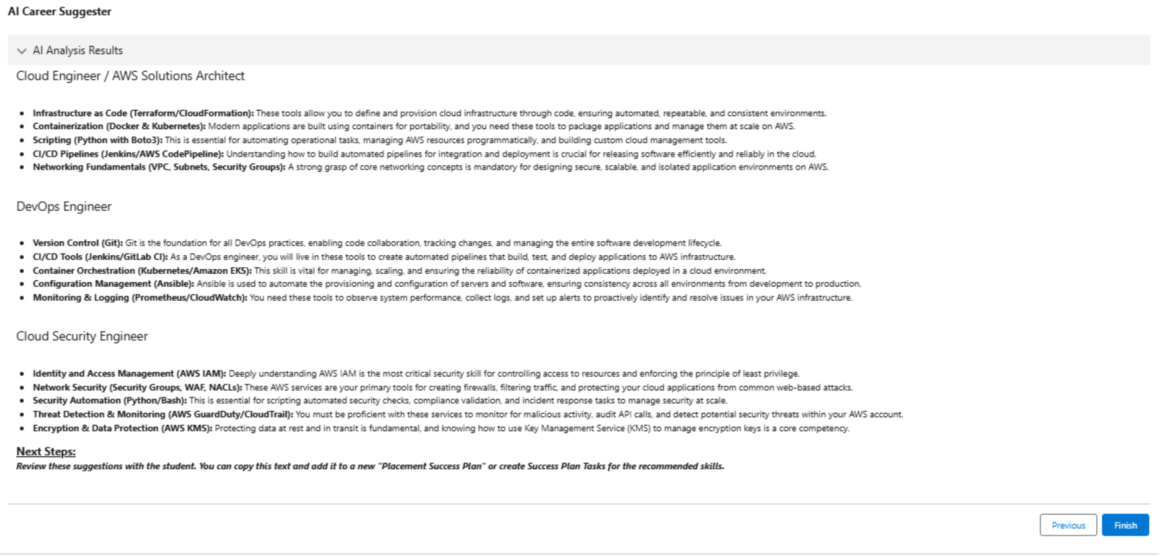
Created By
Sreenidhi Muthyala, 17/10/2025, 7:44 pm

Last Modified By
Sreenidhi Muthyala, 17/10/2025, 7:44 pm

Comments

g/r/00TdM000007POuLUAW/view

5.AI Career Suggestor



AI Career Suggester

Cloud Engineer / AWS Solutions Architect

- **Infrastructure as Code (Terraform/CloudFormation):** These tools allow you to define and provision cloud infrastructure through code, ensuring automated, repeatable, and consistent environments.
- **Containerization (Docker & Kubernetes):** Modern applications are built using containers for portability, and you need these tools to package applications and manage them at scale on AWS.
- **Scripting (Python with Boto3):** This is essential for automating operational tasks, managing AWS resources programmatically, and building custom cloud management tools.
- **CI/CD Pipelines (Jenkins/AWS CodePipeline):** Understanding how to build automated pipelines for integration and deployment is crucial for releasing software efficiently and reliably in the cloud.
- **Networking Fundamentals (VPC, Subnets, Security Groups):** A strong grasp of core networking concepts is mandatory for designing secure, scalable, and isolated application environments on AWS.

DevOps Engineer

- **Version Control (Git):** Git is the foundation for all DevOps practices, enabling code collaboration, tracking changes, and managing the entire software development lifecycle.
- **CI/CD Tools (Jenkins/GitLab CI):** As a DevOps engineer, you will live in these tools to create automated pipelines that build, test, and deploy applications to AWS infrastructure.
- **Container Orchestration (Kubernetes/Amazon EKS):** This skill is vital for managing, scaling, and ensuring the reliability of containerized applications deployed in a cloud environment.
- **Configuration Management (Ansible):** Ansible is used to automate the provisioning and configuration of servers and software, ensuring consistency across all environments from development to production.
- **Monitoring & Logging (Prometheus/CloudWatch):** You need these tools to observe system performance, collect logs, and set up alerts to proactively identify and resolve issues in your AWS infrastructure.

Cloud Security Engineer

- **Identity and Access Management (AWS IAM):** Deeply understanding AWS IAM is the most critical security skill for controlling access to resources and enforcing the principle of least privilege.
- **Network Security (Security Groups, WAF, NACLs):** These AWS services are your primary tools for creating firewalls, filtering traffic, and protecting your cloud applications from common web-based attacks.
- **Security Automation (Python/Bash):** This is essential for scripting automated security checks, compliance validation, and incident response tasks to manage security at scale.
- **Threat Detection & Monitoring (AWS GuardDuty/CloudTrail):** You must be proficient with these services to monitor for malicious activity, audit API calls, and detect potential security threats within your AWS account.
- **Encryption & Data Protection (AWS KMS):** Protecting data at rest and in transit is fundamental, and knowing how to use Key Management Service (KMS) to manage encryption keys is a core competency.

Next Steps:

Review these suggestions with the student. You can copy this text and add it to a new "Placement Success Plan" or create Success Plan Tasks for the recommended skills.