



INSTITUTE OF AERONAUTICAL ENGINEERING (AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

Complex Problem-Solving Self-Assessment Form

1	Name of the Student	K.NAREN
2	Roll Number	25951A66A6
3	Branch and Section	CSE-(AI&ML) – B
4	Program	B. Tech
5	Course Name	FRONT END WEB DEVELOPMENT
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	
EC	Profiles	(✓)
EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
EC 5	Conceptualises alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓
EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	
EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	

	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	Course Outcomes	(✓)
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓
9	Course ELRV Video Lectures Viewed	Number of Videos	Viewing time in Hours
		-	-
10	Justify your understanding of WK1		-
11	Justify your understanding of WK2 – WK9		-
12	How many WKS from WK2 to WK9 were implanted?		-
	Mention them		-

Date:10-12-2025

K.Naren
Signature of the Student

COMPLEX ENGINEERING PROBLEM

**A COURSE SIDE PROJECT
ON
ECOHERO**

KNaren

25951A66A6

ECOHERO

*A Project Report submitted
in partial fulfillment of the
requirements for the award of the degree of*

**Bachelor of Technology
in
CSE (Artificial Intelligence & Machine Learning)**

By

**K.NAREN
25951A66A6**



Department of CSE (Artificial Intelligence & Machine Learning)

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad – 500 043, Telangana

© 2025, K.Naren . All rights reserved.

DECLARATION

I certify that

- a. The work contained in this report is original and has been done by me under the guidance of my supervisor (s).
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute for preparing the report.
- d. I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place: Hyderabad

Date: 10-12-2025

K.Naren
Signature of the Student

CERTIFICATE

This is to certify that the project report entitled **Course Schedule (Topological Sorting)** submitted by K.NAREN to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **CSE - (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** is a Bonafide record of work carried out by his guidance and supervision. The Contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor

Date: 10-12-2025

Head of the Department

Principal

APPROVAL SHEET

This project report entitled **ECOHERO** submitted by *Mr. K.Naren* is approved for the award of the Degree Bachelor of Technology in Branch CSE (**Artificial Intelligence & Machine Learning**).

Examiner

Supervisor(s)

Principal

Date: 10-12-2025

Place: Hyderabad

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Mr.VIDHYASAGAR VIDAPU , Associate Professor, Department of CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** for his kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. Purushotham Reddy, Professor and Head of the Department, Department of CSE (Artificial Intelligence & Machine Learning)**, for extending his support to carry on this project work. I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

I express my sincere gratitude to **Dr. L. V. Narasimha Prasad, Professor and Principal** who has been a great source of information for my work.

I thank our college management and respected **Sri M. Rajashekhar Reddy, Chairman, IARE, Dundigal** for providing me with the necessary infrastructure to conduct the project work.

I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

Abstract

CloudHub is a comprehensive cloud-native platform designed to empower users in creating, organizing, managing, and synchronizing notes seamlessly across multiple devices. Unlike traditional note-taking applications relying on local storage, CloudHub emphasizes real-time collaboration, intelligent organization, robust security, and advanced CRUD functionality.

This project applies Frontend Web Development principles with modern technologies (React.js, Node.js, MongoDB, Firebase) and cloud infrastructure (AWS, Google Cloud) to create a scalable, secure, user-friendly platform. The system enables rich-text note composition, hierarchical organization with notebooks, advanced tagging, version control, and real-time collaboration.

Core features include: user authentication with 2FA, intelligent note CRUD operations with rich text formatting, comprehensive deletion with recovery, version control and revision history, advanced search and filtering, notebook hierarchy, real-time synchronization, collaboration features, analytics dashboards, and end-to-end encryption.

The platform successfully demonstrates how Object-Oriented Programming combined with cloud infrastructure addresses real challenges in digital information management while maintaining data integrity and user privacy.

Comprehensive Report on CloudHub (MuleSoft Anypoint Platform)

Executive Summary

CloudHub, the Integration Platform as a Service (iPaaS) component of the MuleSoft Anypoint Platform, is a fully managed, multi-tenant cloud environment designed for deploying and running Mule applications, APIs, and integrations. Its core value proposition lies in abstracting away infrastructure management, providing global high availability, and enabling secure, scalable connectivity between enterprise systems, whether on-premises or in the cloud. CloudHub's modern architecture, particularly in its latest iteration (CloudHub 2.0), leverages containerization (Kubernetes) to deliver dynamic scaling, improved isolation, and streamlined operations, making it a critical foundation for API-led connectivity strategies.

I. CloudHub Architecture and Deployment Models

1. CloudHub Core Concepts

CloudHub is the runtime plane component of the Anypoint Platform. It consists of the following fundamental building blocks:

- **Platform Services (Control Plane):** These are the shared services that manage the CloudHub environment. They include **Runtime Manager** (the UI/API for deployment and monitoring), **Access Management**, **API Manager**, and **Anypoint Monitoring/Logging**. These services are shared across all tenants but do not process customer data.
- **Workers/Replicas (Data Plane):** These are the dedicated, isolated environments where Mule applications run.
 - **CloudHub 1.0 (Legacy):** Used **Workers** (Virtual Machines) with vCore sizing.
 - **CloudHub 2.0 (Current):** Uses **Replicas** (lightweight containers/pods based on Kubernetes), offering fractional vCore sizing and container-based isolation.
- **Worker/Replica Isolation:** To ensure security, each application runs in its isolated environment (VM or container), preventing custom code from affecting other tenants.

2. Deployment Models (Spaces)

CloudHub 2.0 introduces two primary deployment space types:

Feature	Shared Space	Private Space (Anypoint VPC/VPN)
Isolation	Standard multi-tenant cloud region.	Dedicated, virtual, private, and isolated network space.
Connectivity	Public internet access only.	Connects to private networks (on-premises/private cloud) via Anypoint VPN , Transit Gateway , or AWS Direct Connect .
Load Balancer	Shared CloudHub Load Balancer.	Private Ingress (auto-scaling load balancer) which replaces the old Dedicated Load Balancer (DLB).
Custom Domains	Not directly supported.	Fully supported, allowing vanity URLs and Mutual TLS (mTLS) setup at the Private Space level.
Use Case	Public-facing APIs, integrations without private network access.	Integrations requiring private network connectivity, tighter security, and custom domains.

3. Scalability and High Availability (HA)

CloudHub is designed for mission-critical applications through:

- **Horizontal Scaling:** Deploying an application on **two or more Replicas** (workers) across different Availability Zones (AZs) in a region provides built-in high availability (HA) and load balancing.
- **Automatic Failover:** If a worker or replica fails, CloudHub automatically provisions and migrates the application to a new worker.
- **Clustering:** CloudHub 2.0 supports clustering across replicas, enabling features like in-memory data sharing (Object Store v2) and load distribution.
- **Zero-Downtime Updates:** The platform supports rolling updates, ensuring the old version of an API stays active until the new version is fully deployed and ready to handle traffic.

II. Security and Compliance

Security is a primary feature of the managed CloudHub environment.

1. Data and Runtime Security

- **Encrypted Secrets:** Sensitive configuration data (passwords, certificates) are encrypted at rest and in transit within the platform.
- **Container Isolation:** In CloudHub 2.0, applications run in individual containers, which provide a standardized, secure isolation boundary.
- **OS Patching:** The platform is fully managed, meaning MuleSoft handles continuous operating system patching and updates without requiring customer intervention.
- **Access Management:** Enforced through Anypoint Platform, utilizing role-based access control (RBAC) to limit who can deploy, manage, or access logs for applications.

2. Network Security (Private Spaces)

- **Anypoint VPC/Private Space:** Creates a logical, isolated network. By default, all unallowed traffic is blocked by firewall rules.
- **Outbound Firewall Rules:** Private Spaces allow customers to configure application-level egress rules to control which external endpoints their applications can connect to.
- **TLS/SSL:**
 - **Inbound:** Managed by the Shared Load Balancer or the Private Ingress, supporting TLS v1.2+. Custom certificates can be applied at the Private Space level.
 - **Mutual TLS (mTLS):** Can be enforced at the Private Space to verify the identity of both the client and the server, providing stronger authentication.

3. Compliance

CloudHub adheres to major industry standards, reducing the compliance burden on the customer:

- **ISO 27001:** Information security management.
- **PCI DSS:** For handling credit card data (depending on the customer's implementation).
- **SOC 2:** Controls related to security, availability, processing integrity, confidentiality, and privacy.
- **Regional Data Control:** Allows deployments in specific global regions (US, EU, APAC) to meet data residency requirements.

III. Operations, Monitoring, and Automation

CloudHub simplifies operations by centralizing management through the Anypoint Platform control plane.

1. Deployment and CI/CD

Applications are typically built using **Anypoint Studio** (Mule's IDE) and deployed using one of the following methods:

- **Runtime Manager Console:** Graphical user interface for one-click deployment.
- **Anypoint Platform CLI:** Command-Line Interface for script-based deployments.
- **Mule Maven Plugin:** The standard method for integrating CloudHub deployments into automated **Continuous Integration/Continuous Deployment (CI/CD)** pipelines (e.g., using Jenkins, GitLab CI, GitHub Actions).
- **CloudHub API:** A REST API for advanced automation of deployment and management tasks.

2. Monitoring and Logging

The platform services provide integrated tools for operational visibility:

- **Runtime Manager:** Displays live status, application health, and resource utilization (CPU, memory) for all deployed replicas.
- **Anypoint Monitoring:** Provides advanced dashboards, custom metrics collection, and alerting based on application and platform performance data.
- **Logging Service:** Captures Mule application and deployment logs.
 - Allows log viewing, searching (using a specific query syntax), and downloading.
 - For external integration, logs can be forwarded to external analytics tools (e.g., Splunk, ELK stack) for long-term storage and complex analysis.

3. Worker and Resource Management

- **vCores/Replicas:** CloudHub resources are provisioned in units of vCores (or equivalent replica sizes). Customers choose the appropriate size based on their application's expected workload.
 - **Autoscaling (CloudHub 2.0):** Supports Horizontal Pod Autoscaling (HPA) to dynamically increase or decrease the number of replicas based on CPU utilization or other custom metrics, optimizing cost and performance.
-

IV. Financial and Licensing Model

MuleSoft's pricing is subscription-based, with two main licensing models affecting CloudHub usage:

1. vCore-Based Licensing (Legacy/Capacity)

- Customers purchase a fixed number of **vCores** (virtual processing units) for their subscription term.
- CloudHub applications consume a portion of this purchased vCore capacity.
- **Pros:** Predictable, reserved capacity.
- **Cons:** Risk of overpaying for unused capacity or performance issues if capacity is underestimated.

2. Usage-Based Licensing (Flow/Message)

- This newer model shifts the focus from fixed capacity to actual integration usage.
- Customers purchase bundles of **Mule Flows** (representing the number of distinct integrations deployed) and **Mule Messages** (the volume of transactions processed).
- **Pros:** Cost aligns directly with actual consumption and scales granularly, making it ideal for variable or growing integration needs.
- **Metrics:** Consumption is measured based on the number of integration flows deployed and the volume of messages passing through those flows.

3. Subscription Tiers and Features

CloudHub access and features are tied to the overall Anypoint Platform subscription tier (e.g., Starter, Advanced, Gold, Platinum/Titanium), which dictates access to key capabilities:

- **Basic:** Core features, basic monitoring, and deployment.
- **Advanced:** **Global Multi-Cloud Deployment, Clustering (HA), Advanced Monitoring/Log Management**, and support for **Hybrid Deployment** models.
- **Special Add-ons:** Features like Global Cloud Deployment (multi-region), Dedicated Load Balancer (legacy), and specific support tiers are often sold as add-ons.

V.Code Snippet Examples

A. HTML (Structure)

This is the basic structure (index.html) for a simple webpage, defining the content and linking the external CSS and JavaScript files.

HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CloudHub Demo Page</title>

<link rel="stylesheet" href="styles.css"> </head>

<body>

<header>

<h1>Welcome to CloudHub Deployment</h1>

</header>

<main>

<p>This content is loaded from a simple HTML page.</p>

<button id="actionButton">Click Me</button>

<p id="message"></p>

</main>

<script src="script.js"></script> </body>

</html>
```

B. CSS (Styling)

This CSS (styles.css) provides simple styling for the HTML elements.

CSS

```
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f4f4f9;  
    color: #333;  
}
```

```
header {  
    background-color: #0070c0; /* A MuleSoft/cloud-like blue */  
    color: white;  
    padding: 20px;  
    text-align: center;  
}
```

```
main {  
    width: 80%;  
    margin: 20px auto;  
    padding: 20px;  
    background-color: white;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
}
```

```
#actionButton {  
  padding: 10px 20px;  
  background-color: #f7941d; /* A contrasting orange/yellow */  
  color: white;  
  border: none;  
  cursor: pointer;  
  font-size: 16px;  
}
```

C. JavaScript (Interactivity)

This JavaScript (script.js) adds client-side interactivity by defining a function to run when the button is clicked.

JavaScript

```
// Wait until the entire HTML document is loaded

document.addEventListener('DOMContentLoaded', function() {

    // Get the button element by its ID

    const button = document.getElementById('actionButton');

    // Get the message paragraph element

    const message = document.getElementById('message');

    // Add a click event listener to the button

    button.addEventListener('click', function() {

        // Change the text content of the message paragraph

        message.textContent = 'Action executed! This was handled by client-side JavaScript.';

        // Log to the console for debugging

        console.log('Button clicked and message updated.');

    });

});
```

Date: December 11, 2025

Student Signature: _____

Supervisor Signature: _____

Head of Department Signature: _____