STAC Activity-Based Learning Plan

Software Track (8 Activities)

Goal: Build coding, logic, data handling, and application development skills.

1. Linux Familiarization & Scripting

- **Activity:**
  - Explore Linux commands (`ls`, `grep`, `awk`, `sed`, `cron`).
  - Write a bash script to **auto-organize files** (e.g., move `.jpg` to Images, `.pdf` to Docs).
- **Learning Source:** Linux Journey
- **Outcome:** Confidence in using Linux daily.

2. API Fetching & Mini Dashboard

- **Activity:**
  - Use Python `requests` to fetch real-time data (Weather API or News API).
  - Parse JSON and display results in the terminal.
- **Learning Source:** Real Python – Working with APIs
- **Outcome:** Able to use APIs in projects.

3. Pandas Data Analysis on Real Datasets

- **Activity:**
  - Load a **real dataset** (e.g., World Bank Data or Kaggle COVID-19).
  - Do cleaning, filtering, plotting (`matplotlib`).
- **Learning Source:** Pandas Documentation
- **Outcome:** Learn data wrangling + visualization.

4. Problem Solving (Project Euler)

- **Activity:**
  - Solve 5 problems from Project Euler.
- **Learning Source:** Project Euler
- **Outcome:** Improve logic & algorithmic thinking.

5. OpenCV Fun Challenge

- **Activity:**
  - Use OpenCV to:
    - Detect faces from the webcam.
    - Apply filters (cartoon effect, edge detection).
- **Learning Source:** OpenCV-Python Tutorials
- **Outcome:** Hands-on computer vision skills.

6. Flask Website with API

- **Activity:**
  - Build a Flask app that displays **real-time weather/news** using the API from Activity 2.
- **Learning Source:** Flask Mega-Tutorial
- **Outcome:** First working website with live data.

7. Streamlit Dashboard for Data

- **Activity:**
  - Build a dashboard with Streamlit showing dataset analysis from Activity 3.
- **Learning Source:** Streamlit Docs
- **Outcome:** Learn interactive web-based data apps.

8. Mobile App with Flutter

- **Activity:**
  - Create a Flutter mobile app that:
    - Shows API data (weather/news).
    - Has 2–3 screens.
- **Learning Source:** Flutter Docs
- **Outcome:** Basic cross-platform mobile development.

Hardware Track (8 Activities)

Goal: From Tinkercad basics → robotics simulation with ROS & TurtleBot.

1. Tinkercad Basics

- **Activity:**
  - Simulate a simple LED blink using Arduino.
  - Control the blink with a button.
- **Learning Source:** Tinkercad Circuits
- **Outcome:** Familiarity with Arduino simulation.

2. Sensor Inputs in Tinkercad

- **Activity:**
  - Add LDR sensor (light).
  - Turn the LED ON/OFF depending on brightness.
- **Learning Source:** Tinkercad Arduino examples
- **Outcome:** Sensor → actuator programming.

3. Tinkercad Motor Control

- **Activity:**
  - Control a Servo motor speed using PWM.
  - Use a potentiometer for speed control.
- **Learning Source:** Arduino PWM Basics
- **Outcome:** Hardware logic with motor control.

4. Arduino + LCD Display

- **Activity:**
  - Connect LCD (16x2) and display sensor data (e.g., LDR values).
- **Learning Source:** Arduino LCD Tutorial
- **Outcome:** Display handling in embedded systems.

5. Tinkercad Mini Project – Smart Traffic Light

- **Activity:**
  - Use LEDs + sensors to simulate a **smart traffic system**.
- **Outcome:** First mini project combining multiple components.

6. ROS Basics Setup

- **Activity:**

    - Install ROS2 on Ubuntu.
    - Run publisher-subscriber example (`/chatter`).
- **Learning Source:** ROS2 Tutorials
- **Outcome:** Intro to robotics middleware.

7. Gazebo Simulation (TurtleBot3)

- **Activity:**
    - Launch TurtleBot3 in Gazebo.
    - Use `teleop` to move the robot.
- **Learning Source:** TurtleBot3 eManual
- **Outcome:** Run and control the simulated robot.

8. ROS + OpenCV

- **Activity:**
    - Write a ROS node using OpenCV to process webcam feed.
    - Example: Detect a red-colored object and print its position.
- **Learning Source:** ROS + OpenCV tutorial
- **Outcome:** Robotics + vision integration.
