ASSIGNMENT 5

School of Engineering & Technology

Department: Computer Science & Engineering

Programme: B.Tech CSE (Specialization in AI & ML)

Semester: 1 (Odd)

Course Code: ETCCCP105

Course Name: Computer Science Fundamentals & Career Pathways

Assignment Number: 05

Assignment Title: Design Your Career Roadmap with SMART Goals

Submitted by: Kirtika Khowal

Roll NO.--2501730285

Faculty: Mr.Aryan

Problem definition

Many students miss deadlines or study sessions because they do not have an automated system to schedule study tasks, set reminders, and track completion. The Study Planner with Task Reminder is an automation tool that allows a user to:

- Add study tasks with subject, deadline, priority and estimated time.
- View pending and completed tasks.
- Mark tasks as done.
- Receive reminders (console or email) for imminent deadlines.

Inputs: task title, subject, due date/time, priority (High/Medium/Low), estimated duration.
Outputs: list of tasks (pending/completed), reminder alerts, summary report for the week.

Algorithm for Study planner with task remainder

Initial Setup

START PROGRAM and load any existing TASKS_LIST from storage.

Enter the main program loop.

1. Main Loop & Menu Display

Get the CurrentDate from the system.

Display the main menu options: Add Task, View Schedule, Check Reminders, Mark Task Complete, Exit.

Get UserChoice.

2. Process: Add New Task (If UserChoice is 'Add Task')

Prompt the user to input the following details for the new task:

TaskName

DeadlineDate

Estimated Time (in hours, EstTime)

Priority (1=High, 2=Medium, 3=Low)

Create a new task record with IsComplete set to FALSE.

Add the new task record to the TASKS_LIST.

Sort the entire TASKS_LIST first by DeadlineDate (ascending) and then by Priority (ascending).

Output "Task added and schedule updated."

3. Process: View Daily Schedule (If UserChoice is 'View Schedule')

Initialize an empty SCHEDULE list and TOTAL_STUDY_HOURS to 0.

FOR EACH Task in the TASKS_LIST:

a. IF Task's IsComplete is FALSE:

i. Determine TIME_SLOT (e.g., 2 hours for High Priority, 1 hour for Medium, 0.5 hours for Low).

ii. Set TIME_TO_ALLOCATE to the minimum of TIME_SLOT and the Task's EstTime remaining.

iii. IF TIME_TO_ALLOCATE is greater than 0:

* Add TaskName and TIME_TO_ALLOCATE to the SCHEDULE list.

* Increase TOTAL_STUDY_HOURS by TIME_TO_ALLOCATE.

Display the generated SCHEDULE and the TOTAL_STUDY_HOURS.

4. Process: Check Reminders (If UserChoice is 'Check Reminders')

Initialize REMINDERS_SENT to 0.

FOR EACH Task in the TASKS_LIST:

a. Calculate DaysUntilDeadline (Task.DeadlineDate - CurrentDate).

b. IF Task Priority is High (1) AND DaysUntilDeadline equals 3:

* SEND a high-priority notification.

* Increase REMINDERS_SENT.

c. ELSE IF Task Priority is Medium (2) AND DaysUntilDeadline equals 5:

* SEND a mid-term notification.

* Increase REMINDERS_SENT.

Output the count of REMINDERS_SENT.

5. Process: Mark Task as Complete (If UserChoice is 'Mark Complete')

Prompt the user to input the TaskName to complete.

FOR EACH Task in the TASKS_LIST:

a. IF Task's TaskName matches the input:

* Set Task's IsComplete to TRUE.

* Output confirmation and BREAK the loop.

6. Termination

IF UserChoice is 'Exit Program':

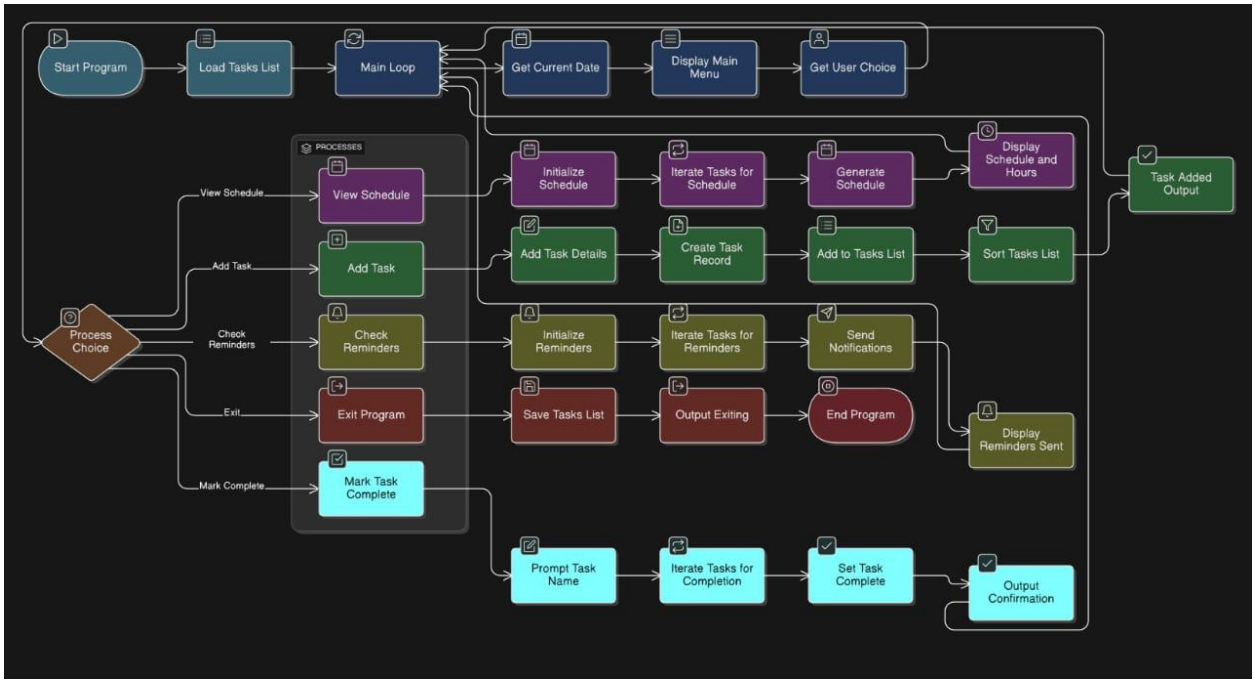
a. Save the current state of TASKS_LIST to persistent storage.

b. Output "Exiting Program."

c. END PROGRAM.

IF UserChoice was not 'Exit Program', loop back to Step 3 to display the menu again

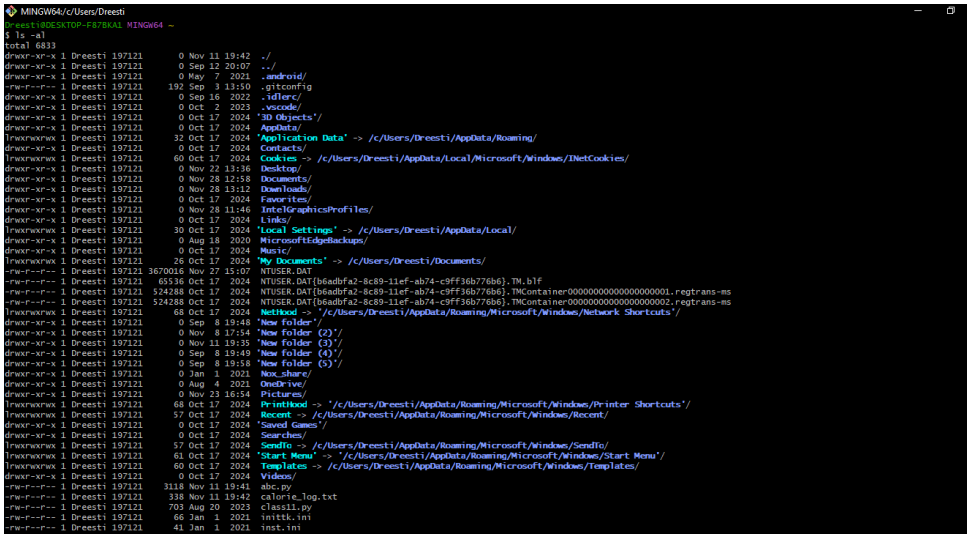
Flowchart


Start Program
Load Tasks List
Main Loop
Get Current Date
Display Main Menu
Get User Choice
PROCESSES
View Schedule
Add Task
Check Reminders
Exit Program
Mark Task Complete
Process Choice
View Schedule
Add Task
Check Reminders
Exit
Mark Complete
Initialize Schedule
Iterate Tasks for Schedule
Generate Schedule
Display Schedule and Hours
Add Task Details
Create Task Record
Add to Tasks List
Sort Tasks List
Initialize Reminders
Iterate Tasks for Reminders
Send Notifications
Display Reminders Sent
Save Tasks List
Output Exiting
End Program
Prompt Task Name
Iterate Tasks for Completion
Set Task Complete
Output Confirmation
Task Added Output

Linux Commnds

1. Pwd

2. Ls –al



3. Mkdir

4. Cd

5. Touch

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~
$ cd Assignment_05

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ touch report_draft.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
```

6. Cp

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ cp report_draft.txt backup.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
```

7. mv

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ mv backup.txt final_report.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
```

8. chmod

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ chmod 740 final_report.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
```

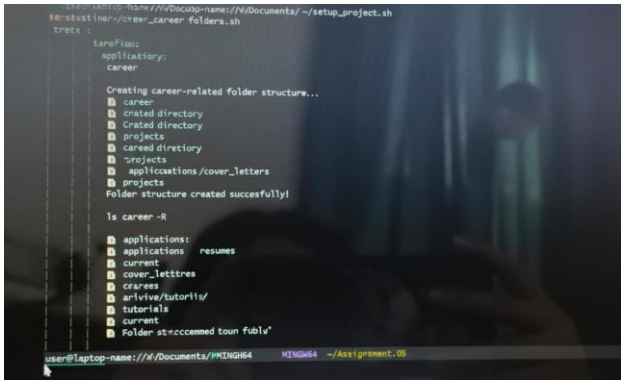
9. ls –l final_report.txt

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ chmod 740 final_report.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ ls -l final_report.txt
-rw-r--r-- 1 Dreesti 197121 0 Nov 28 13:41 final_report.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
```

10. cat

```
Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$ cat report_draft.txt

Dreesti@DESKTOP-F87BKA1 MINGW64 ~/Assignment_05
$
```

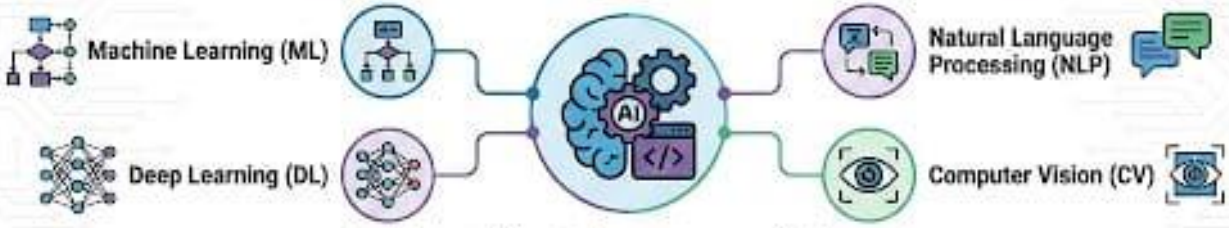
Git-bash



Infographic for Artificial Intelligence

ARTIFICIAL INTELLIGENCE: OVERVIEW, CAREERS, & REAL-WORLD APPLICATION (INDIA)

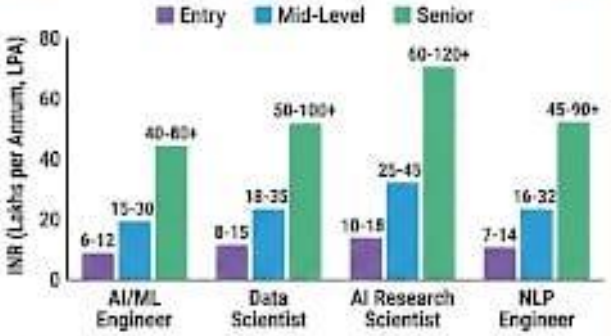
TECHNOLOGY OVERVIEW



Enabling machines to learn, reason, and self-correct.

Core Functions: Data Ingestion -> Processing & Algorithms -> Model Training -> Inference & Action

JOB ROLES & SALARY TRENDS



Salary Trends: Increasing demand, high growth potential

CERTIFICATIONS (LINK TO TASK 4)

- Google Cloud Professional Machine Learning Engineer
- AWS Certified Machine Learning – Specialty
- Microsoft Certified: Azure AI Engineer Associate
- DeepLearning.AI TensorFlow Developer Professional Certificate
- Stanford University Machine Learning (Coursera)

Industry-recognized credentials validating AI expertise.

INDIAN STARTUP EXAMPLE: NIRAMAI HEALTH ANALYTICS

Niramai's AI-driven thermal analytics for early breast cancer detection

Using AI & ML to develop a novel, non-invasive, radiation-free, and privacy-aware breast cancer screening solution. Transforming healthcare accessibility in India.

Career planning and professional readiness

Smart goals

Short-Term Skill Acquisition (6 Months) S: Successfully complete the foundational courses and a practical project (e.g., a simple image classifier) using Python, TensorFlow, and essential libraries like NumPy/Pandas. M: Project repository uploaded to GitHub by (Month/Year - e.g., March 2026) with a README, and a minimum grade of 85% in the relevant semester courses.

Medium-Term Industry Readiness (18 Months) S: Secure and successfully complete a 6-week summer internship or a relevant industry project focused on Robotics simulation (e.g., using ROS - Robot Operating System). M: Internship completion certificate received by (Month/Year - e.g., June 2027), a positive evaluation report, and a detailed project presentation delivered.

Long-Term Career Launch (4 Years) S: Secure a full-time position as a Junior AI Engineer or Robotics Software Developer at a reputable technology firm (e.g., in the areas of computer vision or autonomous systems). M: Employment offer letter received within six months of graduation (by (Month/Year - e.g., July 2029)), with a salary package meeting industry standards for the role.

Certification

Microsoft Certified: Azure AI Fundamentals (AI-900) Microsoft / Azure Validates core AI concepts, machine learning workloads, and essential cloud services (Azure), which are crucial for deploying scalable AI models in enterprise environments. 1-2 Months Study, ~$99 USD Exam Fee

TensorFlow Developer Certificate Google / TensorFlow Confirms proficiency in building and training Machine Learning models using TensorFlow, the leading open-source framework for deep learning and neural networks. Essential for core AI development. 2-3 Months Study, ~$100 USD Exam Fee

ROS (Robot Operating System) Certification Multiple (e.g., The Construct, official training partners) Though not a single official body, proficiency certificates in ROS are vital for robotics engineers. ROS is the middleware framework for creating complex robot applications. 3-4 Months Study, Varies based on course chosen.

Career roadmap

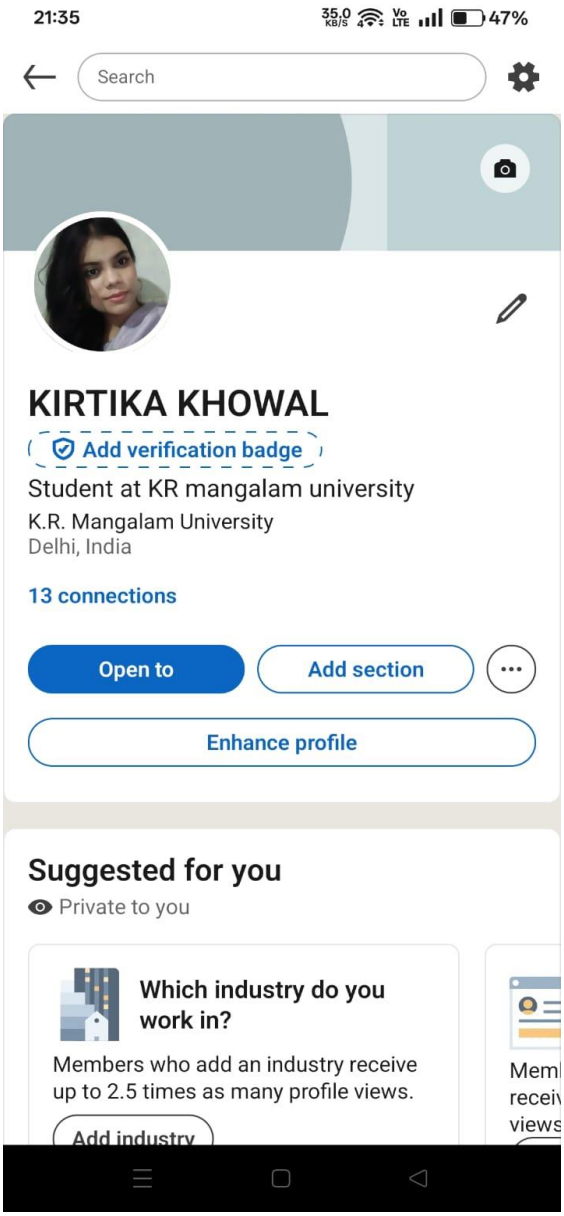
Phase 1: Foundations (Year 1) 12 Months Focus: Data Structures & Algorithms (DSA), Core Programming (Python/C++), Discrete Mathematics, Introduction to Robotics and AI concepts. Milestones: Strong grasp of DSA, completion of two foundational Python projects, Active GitHub profile, LinkedIn Profile Screenshot (required for the assignment).

Phase 2: Specialization (Year 2) 12 Months Focus: Machine Learning (ML) Theory, Deep Learning (DL), Cloud Basics (Azure/AWS), Begin TensorFlow certification study. Milestones: Completion of AI-900 Certification, secure a small ML-focused project using a real dataset, begin open-source contributions.

Phase 3: Real-World Application (Year 3) 12 Months Focus: Robotics (ROS), Computer Vision, Internship/Major Industry Project, Participation in a national hackathon. Milestones: Complete 6-week summer internship, proof of hackathon participation/plan (required for the assignment), Advanced project portfolio for Robotics/Vision.

Phase 4: Placement & Launch (Year 4) 12 Months Focus: Technical interview preparation (DSA revision, System Design), Final Year Project, Job application submission, Networking. Milestones: Successful completion of the Final Year Project, achieve long-term SMART goal (securing a Junior AI/Robotics role).

Linkedin profile



Hackathon Participation Plan: Robotics & AI Challenge

To demonstrate applied skills and collaboration, the medium-term career roadmap includes participation in a significant technical competition.

Chosen Hackathon Target

The target competition is a Robotics Simulation Challenge (e.g., a university-hosted event or a major industry hackathon like Smart India Hackathon or similar focused challenges).

Project Concept: Autonomous Warehouse Navigation

Goal: Develop an AI-driven script for a simulated robotic arm and mobile base (using ROS and Gazebo simulation environments) to efficiently navigate a virtual warehouse, identify objects via computer vision (OpenCV/YOLO), and move them to designated storage locations.

Technologies:

Robot Operating System (ROS): For communication and control of simulated components.

Python: Main programming language for logic and control.

OpenCV / TensorFlow Lite: For visual detection and object recognition.

Gazebo: The 3D robotics simulator environment.

Team Role: AI & Vision Specialist. My primary contribution will be developing the robust computer vision pipeline that processes the simulated camera feed to accurately locate and classify the required objects for manipulation by the arm.

Success Metric: Completing a defined number of picking and placement tasks within a 10-minute time limit in the simulation environment.

Reflection

The process of completing this assignment has been a foundational exercise in both technical execution and strategic career planning. It provided a necessary bridge between theoretical knowledge and applied industry readiness.

One of the primary challenges faced was the troubleshooting of the Git Bash script execution. The unexpected interaction between Windows (CRLF) and Unix (LF) line endings required a deep dive into fundamental shell behavior and file formatting. This experience underscored a crucial lesson: that successful software deployment depends not just on correct code syntax, but on understanding the underlying operating system environment. The iterative debugging process significantly improved my problem-solving skills and resilience when facing ambiguous errors.

Another key skill improved was structured career forecasting. Defining SMART goals forced a tangible and measurable plan for skill acquisition, moving beyond vague aspirations to actionable steps like obtaining the Azure AI Fundamentals certification by a specific deadline. The construction of the career roadmap clearly visualizes the transition from academic theory to industry application, ensuring my B.Tech specialization remains relevant and highly focused.

I will apply this learning by rigidly adhering to the defined roadmap, prioritizing the acquisition of skills like TensorFlow and ROS proficiency during the next academic year. The disciplined approach to using open-source tools, as practiced with Git Bash and the project setup script, will be a permanent fixture in my future professional workflow, ensuring every project I undertake begins with a standardized, automated, and scalable structure, preparing me directly for a role as a Junior AI Engineer.