internStudio

ARTIFICIAL INTELLIGENCE INTERNSHIP

TASK GUIDE

Build systems that reason. Perception, search, language and decision-making.

SIX ADVANCED PROJECTS — CHOOSE ANY TWO

01 **Intelligent Document Question-Answering System**

02 **Conversational Assistant with Tool Use**

03 **Real-Time Object Detection & Counting System**

04 **AI-Powered Resume Screening & Ranking System**

05 **Search-Based Game-Playing Agent**

06 **Speech-to-Text Meeting Summarizer**

INTERNSHIP
Artificial Intelligence

REQUIRED TASKS
Complete any 2 projects

FOCUS
NLP · Computer Vision · Search & Planning · Intelligent Agents

INTERNSHIP INSTRUCTIONS

Your Internship Tasks

Select and complete **any 2 projects** from the task list. AI is broader than model training — search, planning, perception and language all count, as long as the system genuinely reasons over its input.

Define	Design	Build	Evaluate	Refine	Deploy

EACH PROJECT SHOULD DEMONSTRATE

- [ ] A clearly defined problem the system solves
- [ ] An appropriate AI technique for that problem
- [ ] Proper handling of the input modality (text, image, audio, state space)
- [ ] A real evaluation, not just "it seems to work"
- [ ] Analysis of failure cases
- [ ] Reasonable performance under realistic conditions
- [ ] A usable interface to interact with the system
- [ ] Honest discussion of the system's limitations

EVERY SUBMISSION INCLUDES

- [ ] GitHub repository
- [ ] Source code or notebooks
- [ ] README explaining the approach
- [ ] Evaluation results and metrics used
- [ ] Sample inputs and outputs / a short demo
- [ ] Requirements / dependency file
- [ ] Screenshots or a demo link where applicable

> **Suggested technologies:** Python · PyTorch or TensorFlow · Hugging Face Transformers · OpenCV · spaCy/NLTK. Optional: LangChain, an LLM API, a game/search library.

Submit your completed work here

Once both projects are evaluated, documented and pushed to GitHub, submit your project details through the official submission form.

https://forms.gle/vxL5JE8cvRXXfDBv8

> **A note on quality.** "It gives a plausible-looking answer sometimes" is not a result. Evaluate against real cases, show where it fails, and explain why.

01 Intelligent Document Question-Answering System

Build a system that answers natural-language questions by retrieving and reasoning over a real document collection.

REQUIREMENTS

- Ingest and chunk a real document collection
- Generate answers grounded in retrieved content
- Handle questions with no good answer in the corpus
- Measure retrieval precision, not just answer fluency
- Build a retrieval mechanism (embeddings or search)
- Cite the source passage for each answer
- Evaluate answer quality on a held-out question set
- Provide a simple query interface

Focus: Retrieval-augmented reasoning, grounding, and evaluation of a language system.

02 Conversational Assistant with Tool Use

Build an assistant for a specific domain (e.g. travel planning, scheduling) that can call real tools/functions to complete tasks.

REQUIREMENTS

- Define a clear domain and set of callable tools
- Handle multi-turn conversations with context
- Handle ambiguous or missing information gracefully
- Evaluate task-completion rate on test scenarios
- Route user intent to the correct tool call
- Validate and confirm actions before executing them
- Log tool calls and their outcomes
- Provide a usable chat interface

Focus: Tool-augmented reasoning, multi-turn dialogue management and task-completion evaluation.

03 Real-Time Object Detection & Counting System

Build a computer vision system that detects and counts objects of interest in a video stream (e.g. vehicles, people).

REQUIREMENTS

- Real-time or near-real-time detection on video
- Count unique objects crossing a defined zone/line
- Report detection confidence, not just raw counts
- Visualize detections and counts on the video
- Track objects across frames, not just detect per frame
- Handle occlusion and overlapping objects reasonably
- Evaluate accuracy against manually verified ground truth
- Discuss failure modes (lighting, angle, crowding)

Focus: Object detection, multi-object tracking and quantitative accuracy evaluation.

04 AI-Powered Resume Screening & Ranking System

Build a system that ranks candidate resumes against a job description using NLP.

REQUIREMENTS

- Parse resumes into structured information
- Extract and match skills against the job description
- Score and rank candidates with an explainable method
- Handle varied resume formats robustly
- Surface why a candidate was ranked where they were
- Evaluate ranking quality against human judgments
- Flag potential bias in the ranking approach
- Provide a simple review interface for recruiters

Focus: NLP-based information extraction, explainable ranking and bias-aware evaluation.

05 Search-Based Game-Playing Agent

Build an agent that plays a non-trivial board or puzzle game using search and heuristics, not supervised learning.

REQUIREMENTS

- Implement a proper game-state representation
- Implement a search algorithm (minimax, alpha-beta, MCTS)
- Design a meaningful heuristic evaluation function
- Set a configurable search depth / time budget
- Benchmark the agent against a random and a greedy baseline
- Measure win rate and decision time
- Handle edge cases (draws, terminal states) correctly
- Provide a way to play against the agent

Focus: Search algorithms, heuristic design and rigorous agent-vs-agent evaluation.

06 Speech-to-Text Meeting Summarizer

Build a pipeline that transcribes spoken audio and produces a structured, accurate meeting summary.

REQUIREMENTS

- Transcribe real audio to text
- Handle multiple speakers or at least speaker turns
- Clean and segment the transcript into topics
- Generate a structured summary (decisions, action items)
- Evaluate transcription accuracy (WER or similar)
- Evaluate summary quality against a human-written reference
- Handle noisy or low-quality audio gracefully
- Provide the transcript and summary as exportable output

Focus: Speech processing, summarization, and evaluation against both accuracy and quality baselines.

FINAL STEP

Ready to Submit?

Verify everything below before you send your work in.

SUBMISSION

- [ ] GitHub repository
- [ ] README explaining the approach
- [ ] Sample inputs/outputs or demo
- [ ] Requirements / dependency file
- [ ] Submission form completed
- [ ] LinkedIn post where required

SYSTEM

- [ ] Usable interface or demo
- [ ] Reasonable runtime performance
- [ ] Edge cases considered

PROBLEM

- [ ] Problem clearly scoped
- [ ] Appropriate AI technique selected
- [ ] Input modality handled correctly

EVALUATION

- [ ] Real evaluation metric used
- [ ] Failure cases analyzed
- [ ] Performance discussed honestly

Important. Complete any 2 projects. You should be able to explain why you chose your approach, how you evaluated it, and where it breaks down.

Submission Form

Submit your repository, live link and screenshots here.

https://forms.gle/vxL5JE8cvRXXfDBv8

internStudio

LEARN · BUILD · GROW

www.internstudio.in

team.internstudio@gmail.com