YOUR PROJECT TITLE GOES HERE

Your Name (Roll Number: XXXXX)

Internship / Term Project Report
Trimester 7 / 8 / 9

B.Sc. (Honours) Data Science and Artificial Intelligence
Indian Institute of Technology Guwahati, India
your-email@op.iitg.ac.in

ABSTRACT

The abstract should provide a concise summary of the entire project. It must clearly answer the following questions: (1) What problem did you work on? (2) Why is the problem important? (3) What approach did you take? (4) What were the key outcomes or insights?

Good practice:

- Write the abstract *after* completing the report
- Keep it factual and precise (no storytelling here)
- Limit to 150–200 words

Example: This report presents the work carried out during a three-month internship on *[problem domain]*. The objective was to *[objective]*. A *[method/approach]* was developed and evaluated using *[data/tools]*. Experimental results indicate that *[key outcome or insight]*.

***Index Terms*—** List 4–6 keywords that best describe your work (e.g., Machine Learning, Computer Vision, Optimization).

1. INTRODUCTION

The introduction sets the context for your work. Assume that the reader is technically trained but not familiar with your specific project.

This section should:

- Introduce the application or problem domain
- Explain why the problem matters (practical or research relevance)
- Briefly summarize what you did

Good practice:

- Avoid detailed technical explanations here
- Do not include results or conclusions
- End with a short paragraph describing the structure of the report

Example closing sentence: Section 2 describes the problem formulation and objectives. Section 3 details the methodology. Experimental results are presented in Section 4, followed by discussion and conclusions.

2. PROBLEM STATEMENT AND OBJECTIVES

2.1. Problem Statement

Clearly and precisely define the problem you worked on. A reader should be able to understand the problem without reading the rest of the report.

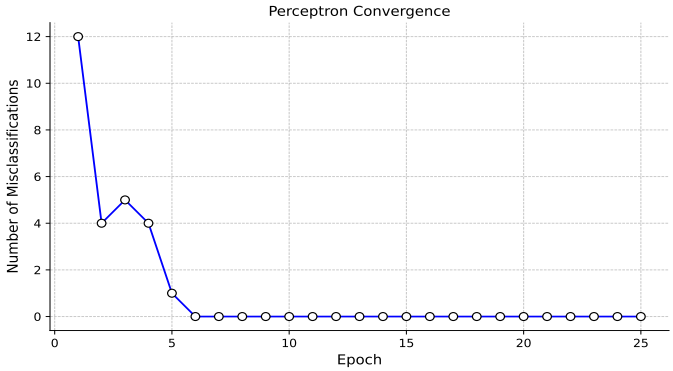

Fig. 1. Illustration of perceptron convergence for a training set comprising of 100 sample points drawn from two classes (balanced).

Good practice:

- Avoid vague statements such as “we worked on AI”
- Specify constraints, assumptions, and scope
- Mention real-world relevance if applicable

2.2. Objectives

List the specific goals you aimed to achieve during the project.

Good practice:

- Objectives should be measurable or verifiable
- Use action verbs (design, analyze, evaluate, implement)

Example:

1. To study existing approaches for *[problem]*
2. To design and implement a *[model/system]*
3. To evaluate performance using appropriate metrics

3. METHODOLOGY / APPROACH

This section explains *how* you addressed the problem. It should be detailed enough for another student to reproduce your work.

3.1. Overall Workflow

Provide a high-level overview before diving into details. You can include figures in your report for this (see Figure 1).

Good practice:

- Start from data or inputs and end with outputs
- Explain each block briefly in the text

3.2. Technical Details

Describe algorithms, models, tools, or frameworks used.

Good practice:

- Clearly justify design choices
- Reference prior work where applicable
- Include equations only when they add clarity

Example: The loss function used for training is given by

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, f(\mathbf{x}_i)), \tag{1}$$

where $\ell(\cdot)$ denotes the task-specific loss.

4. EXPERIMENTS AND RESULTS

This section presents the experimental setup and results objectively.

4.1. Experimental Setup

Describe datasets, tools, and evaluation protocols.

Good practice:

- Mention dataset size and splits
- Specify hardware and software environment
- Clearly define evaluation metrics

4.2. Results

Present results using tables and figures.

Good practice:

- Refer to every table and figure in the text
- Do not interpret results here (save that for Discussion)

5. DISCUSSION

Interpret and analyze the results presented earlier.

This section should address:

- Why certain methods performed better or worse
- Trade-offs observed during experimentation
- Limitations of the approach

Good practice:

- Be honest about failures or challenges
- Connect observations back to objectives

6. CONCLUSION AND FUTURE WORK

Summarize the work and reflect on learning outcomes.

Good practice:

- Do not repeat the abstract
- Highlight key contributions and insights
- Suggest realistic future improvements

Example future work: Future extensions could include exploring larger datasets, alternative models, or real-time deployment.

Note to students: Clear scientific writing is an essential research skill. Students are encouraged to reflect on their learning process and presentation quality, and may consult well-known resources on effective research writing and paper structure (e.g., [1, 2, 3]).

7. ARTIFACTS AND DEMONSTRATIONS

This section provides links to external resources that complement the report. Students are encouraged to share reproducible artifacts that demonstrate their implementation, experiments, or system behavior.

Examples of acceptable artifacts include:

- Source code repositories (e.g., GitHub, GitLab)
- Technical blogs or project webpages
- Video demonstrations (e.g., YouTube, institutional cloud links)
- Interactive demos or dashboards (if applicable)

Good practice:

- Ensure links are publicly accessible or shared with appropriate permissions
- Clearly describe what each link contains
- Avoid linking raw folders without explanation
- Ensure content reflects your own work and contributions

Example:

- **Code Repository:** `https://github.com/username/project-name`
- **Project Blog:** `https://medium.com/@username/project-summary`
- **Video Demo:** `https://youtu.be/xxxxxx`

Note: Use of AI-assisted tools (e.g., ChatGPT, Gemini, Copilot) is permitted to enhance understanding and productivity. Students must ensure originality, proper attribution, and a clear understanding of all submitted work.

8. REFERENCES

[1] George D. Gopen and Judith A. Swan, "The science of scientific writing," *American Scientist*, vol. 78, no. 6, pp. 550–558, 1990.

[2] Donald E. Knuth, "Literate programming," *The Computer Journal*, vol. 27, no. 2, pp. 97–111, 1984.

[3] Simon Peyton Jones, "How to write a great research paper," *Microsoft Research*, 2003.