Live Project Documentation & Source Code Repository

Team Assignment Document

---

1. Objective

The objective of this assignment is to create a **comprehensive repository of 500 live projects** for the company website. These projects will be used for **learning, training, demonstrations, and commercial purposes**. Each project must be **fully documented, user-focused, and supported with downloadable source code**.

---

2. Domains & Project Allocation

The projects must be developed and documented under the following five domains:

Domain	Number of Projects
Web Development	100
Artificial Intelligence	100
Machine Learning	100
Data Science	100
Cybersecurity	100
Total	**500 Projects**

---

3. Mandatory Project Documentation Structure

Each project **must strictly follow** the documentation format below. Any deviation will require rework.

3.1 Project Title

- Must be clear, descriptive, and professional
- Should immediately convey the project purpose
- Avoid vague or generic naming

3.2 Project Introduction

- Overview of the project
- Real-world problem addressed

- Target users or industry relevance

Length: 150–250 words
Focus: Clarity, usefulness, and user engagement

3.3 Implementation / Framework

This section must explain:

- Overall system workflow or architecture
- Core logic, algorithm, or methodology
- Frameworks, libraries, and APIs used

The explanation should be **technical yet easy to understand**.

3.4 Output Screenshot

- Minimum **3 screenshots per project**
- Screenshots must show actual working output
- UI images must be clean, readable, and user-oriented

Formats: .png or .jpg

3.5 Technical Skills Used

Clearly list:

- Programming languages
- Concepts
- Algorithms
- Libraries / Frameworks

3.6 Tools Used

Mention all relevant tools, such as:

- IDEs
- Platforms / Operating Systems
- Databases
- Third-party tools

3.7 Source Code (Mandatory)

Each project must include:

- Fully working and tested source code - downloadable ZIP file

Source code must contain:

- README.md with setup and execution steps
- Dependency file (requirements.txt, package.json, etc.)
- Clear run instructions

3.8 Project Metadata

Field	Description
Domain	One of the five assigned domains
Difficulty Level	Beginner / Intermediate / Advanced
Duration	Estimated completion time

---

4. User Attention & Usability Guidelines (Mandatory)

All projects must be developed and documented with **end users in mind**.

- Documentation must be **simple, structured, and easy to follow**
- Use headings, bullet points, and step-by-step explanations
- Screenshots should help users quickly understand the output
- README files must be beginner-friendly
- Avoid unnecessary complexity in UI or logic
- Projects should be **educational, practical, and demo-ready**

---

5. Quality Standards

✓ Code must execute without critical errors
✓ Documentation must be professional and well-organized
✓ Naming conventions must be consistent
✓ Output screenshots must match the code behavior.

---

6. Folder Structure (Per Project)

project-name/

│

├── documentation.md

├── source-code/

├── screenshots/

├── README.md

└── requirements.txt

---

7. Domain-Specific Expectations

Web Development

- Responsive and user-friendly UI
- Proper navigation and validation

Artificial Intelligence

- Clear explanation of AI behaviour and results

Machine Learning

- Dataset explanation
- Model training and evaluation details

Data Science

- Data analysis with meaningful visualizations

Cybersecurity

- Ethical and legal use cases only
- Clear explanation of vulnerabilities and mitigation

---

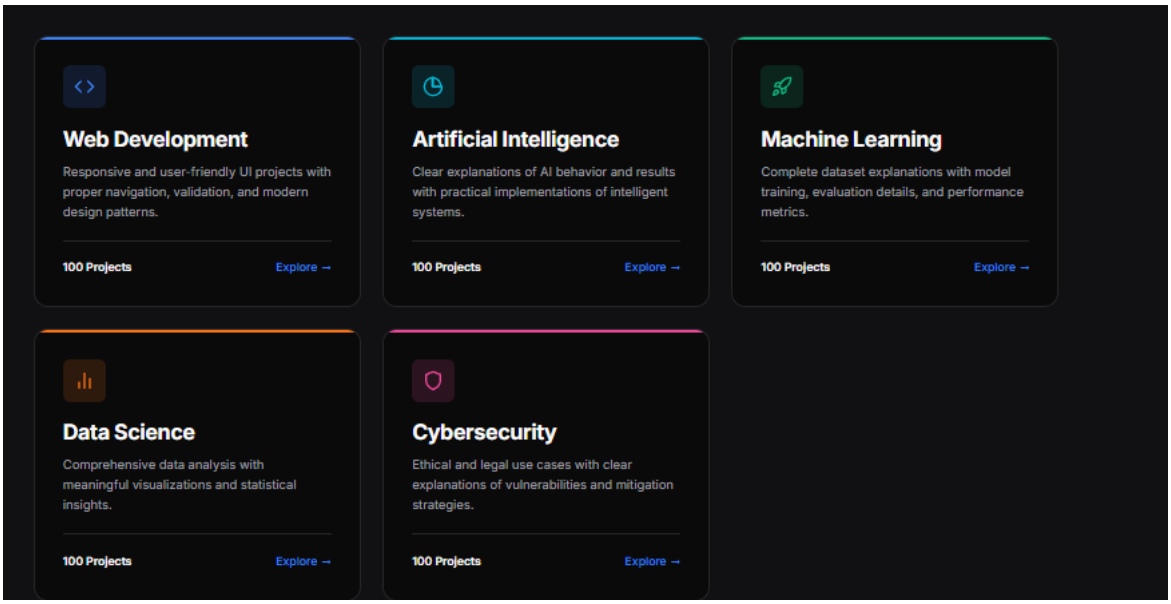
8. Delivery & Review Process

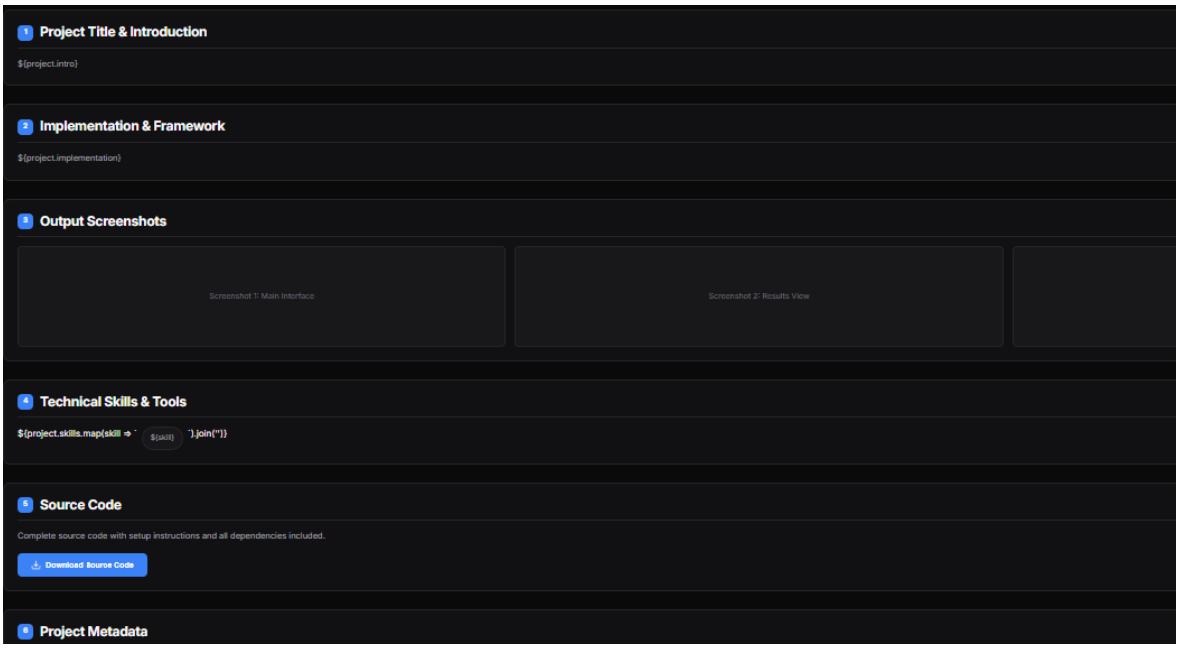
- Projects must be delivered **domain-wise**
- Each project must be independently downloadable
- Internal QA review is mandatory
- Revisions must be completed based on feedback

---

9. Final Note

This repository represents the organization’s technical standards.
User attention, clarity, usability, and code quality are equally important.
Production-level professionalism is expected in all deliverables.





Production-Level Standards

All projects meet strict quality criteria ensuring professional, reliable, and educational content.

- **Original Content**
  100% plagiarism-free, original code and documentation.
- **Tested & Verified**
  Code executes without critical errors on standard environments.
- **Professional Documentation**
  Well-organized, beginner-friendly explanations and guides.
- **Consistent Naming**
  Standardized naming conventions across all project files.
- **Accurate Screenshots**
  Output images match the actual code behavior precisely.
- **User-Focused Design**
  Simple, structured content with clear step-by-step guidance.

Mandatory Project Structure

Every project follows a strict documentation format ensuring consistency and quality across all deliverables.

```
project-name/
|
|-- documentation.md
|-- source-code/
|-- screenshots/
|-- README.md
|-- requirements.txt
```

1. **Project Title & Introduction**
   Clear, descriptive naming with 150-250 word overview addressing real-world problems.
2. **Implementation & Framework**
   System workflow, core logic, algorithms, and technical stack explanation.
3. **Output Screenshots**
   1-3 clean, readable screenshots showing actual working output.
4. **Technical Skills & Tools**
   Complete list of languages, libraries, frameworks, IDEs, and platforms used.
5. **Source Code**
   Fully working, tested code with setup instructions and dependencies.
6. **Project Metadata**
   Domain, difficulty level, estimated duration, and version information.