# Workshop Project Report: CogniForm – Interactive Cognitive Science Survey Platform

## 1. Project Description

CogniForm is an open-source, modular survey platform designed to deliver and analyze interactive cognitive science tasks. The platform enables researchers and students to define psychological tasks as object-oriented components, expose them through a web API, and collect, validate, and analyze user data through a structured data pipeline.

CogniForm aims to bridge cognitive behavioral research and modern collaborative software practices. Participants will create a working prototype that integrates survey design, API development, data storage, visualization, and packaging, all while working in teams with Git, Docker, and CI/CD workflows.

## 2. Project Objectives and Sub-objectives

- **Primary Objective:** Design and implement a modular, Dockerized Python platform for interactive cognitive surveys with end-to-end data handling and analysis.
- **Sub-objectives:**
  - Design object-oriented classes for survey tasks such as the Cognitive Reflection Test, Stroop task, and Delay Discounting.
  - Build a RESTful API using FastAPI for task presentation and data collection.
  - Implement a data pipeline to validate, transform, store, and analyze responses.
  - Visualize survey results using Matplotlib and Seaborn.
  - Package the system with Docker and provide structured documentation.
  - Collaborate using Git and follow best practices in clean code, testing, and CI.

## 3. Workshop Learning Objectives and Sub-objectives

- **Programming and Tools:**
  - Review of Python and Linux/PowerShell
  - Object-Oriented Programming (OOP) and modular design

  - Use of virtual environments and clean code principles

- **Collaboration and Git:**
  - Solo and team-based Git workflows (branches, PRs, merge conflicts)
  - CI/CD pipelines and version control best practices
  - GitHub project management and open-source workflows

- **Project Design and Management:**
  - SMART goal setting and WBS (Work Breakdown Structure)
  - Gantt chart for time management
  - Team-building and feedback techniques (retrospectives)

- **Scientific Computing:**
  - Use of scientific libraries: Numpy, Scipy, Pandas, Matplotlib
  - Automated testing and documentation techniques
  - Data visualization principles and common mistakes

- **Deployment and Packaging:**
  - Docker basics and containerized deployment
  - Project packaging using `setuptools`
  - CI/CD and documentation publishing

# 4. Prerequisites for Participants

- Familiarity with basic Python (functions, classes, file I/O)
- Awareness of Git basics (clone, commit, push)
- Experience with command-line interfaces (Linux or PowerShell)
- Optional: Knowledge of JSON, REST APIs, Markdown

# 5. Step-by-Step Plan

1. **Project Kickoff & Planning**
   - Define SMART goals and create WBS.
   - Assign team roles (lead, backend, frontend, analyst, tester).
2. **Tooling Setup**

- Set up Git repository and branching strategy.
- Create a Python virtual environment.
- Initialize project structure and README.

3. **Survey Design**
   - Design survey logic and flow (CRT, Stroop, Delay Discounting).
   - Implement task classes using OOP.
4. **API Development**
   - Create endpoints using FastAPI.
   - Handle user session, task dispatch, and response collection.
5. **Data Pipeline & Analysis**
   - Validate and clean response data.
   - Store in structured format (e.g., SQLite or JSON).
   - Visualize results with Matplotlib/Seaborn.
6. **Packaging & Deployment**
   - Create Dockerfile and docker-compose config.
   - Add setup.py for packaging and documentation.
   - Implement basic CI for testing and linting.
7. **Documentation & Final Presentation**
   - Generate user manual and developer guide.
   - Present workflow, challenges, and results.

# 6. Expected Deliverables

- Git repository with modular codebase
- Interactive RESTful cognitive survey
- Python classes for each task (CRT, Stroop, Delay Discounting)
- FastAPI application with endpoints for task and user session handling
- Dockerized deployment setup
- Data transformation and analysis pipeline
- Visualizations of user behavior and task performance
- Test suite and CI workflow
- Complete structured documentation

# 7. Cognitive Science Survey Details

## Section 1: Demographics

- Age, gender (optional), education level
- Familiarity with cognitive science (1–5 scale)

## Section 2: Cognitive Reflection Task (CRT)

**Purpose:** Measure reflection vs intuition. **Questions:**

1. A bat and a ball cost $1.10 in total. The bat costs $1.00 more than the ball. How much does the ball cost?
2. If it takes 5 machines 5 minutes to make 5 widgets, how long would it take 100 machines to make 100 widgets?
3. In a lake, a patch of lily pads doubles in size each day. If it takes 48 days to cover the lake, how long does it take to cover half the lake?

**Metrics:** User response, edits before submit, response time

## Section 3: Stroop-like Task

**Purpose:** Test attentional control and inhibition. **Mechanism:** Users must identify the font color of incongruent color words. **Example:** Word "Blue" shown in red font → Answer: Red **Trials:** 10 randomized **Metrics:** Accuracy, response time per trial

## Section 4: Delay Discounting

**Purpose:** Assess impulsivity and preference for delayed reward. **Example:** Choose between:

- Receive $10 today
- Receive $15 in a week
- Receive $20 in a month

**Metrics:** Delay preference, estimated discount rate

## Section 5: Consistency Check

- Slightly altered repetition of 1–2 previous questions
- Measure intra-survey consistency

## Data Collected

- Participant metadata (ID, timestamp, demographics)
- Task-specific responses
- Accuracy and response time logs
- Inferred behavioral indices (e.g., reflection, impulsivity)