Programming CW2 – Project Plan

**Phase 1: Initial Planning and Setup (Week 1: 17th–23rd March)**

* **Kick-off meeting (whole team)**
  + Discuss and agree on a specific PI concept relevant to university students.
  + Define roles clearly for accountability (Scrum Master, Product Owner, Researchers, Designers, Developers, Testers, Documenters).
  + Set up communication channels (Slack, Discord, WhatsApp) and management tools (Trello, Jira, GitHub).
* **Recommended role distribution:**
  + **Scrum Master (1 person)**: Facilitates meetings, ensures Agile methodology adherence.
  + **Product Owner (1 person)**: Defines product vision, prioritizes requirements, oversees user research.
  + **Researchers (2 persons)**: Conduct literature review, user interviews, analyze existing PI solutions.
  + **Designers (2 persons)**: Prepare UML diagrams and interface designs.
  + **Developers (2 persons)**: Implement software (mock-ups, prototype, functionalities).
  + **Test Lead (1 person)**: Creates testing plans, manages test execution, and documentation.

**Phase 2: Research and Requirements Gathering (Weeks 2–3: 24th March–6th April)**

* **Researchers & Product Owner:**
  + Conduct background research, review 6+ articles, with at least 3 peer-reviewed sources.
  + Conduct user interviews (students) to gather feedback on PI concept.
  + Document research findings clearly to guide development and design.
* **Scrum Master:**
  + Schedule and facilitate weekly Scrum meetings.
  + Document sprint backlog, tasks, and meeting minutes clearly.

**Phase 3: Sprint 1 – Specification and Initial Development (Weeks 4–6: 7th–27th April)**

* **Sprint Planning Meeting (whole team)**:
  + Agree sprint goals: finalizing software requirements specification and initial prototype development.
* **Designers & Developers:**
  + Create detailed UML diagrams and user interface sketches.
  + Implement basic functionalities (data collection, manual input, basic UI).
* **Test Lead:**
  + Draft initial test plans based on specifications.
  + Carry out unit testing and document results.
* **Sprint Review & Retrospective Meeting** (end of week 6):
  + Present initial prototype to tutors.
  + Identify issues, document feedback, and adjust backlog accordingly.

**Phase 4: Sprint 2 – Implementation and Comprehensive Testing (Weeks 7–8: 28th April–9th May)**

* **Sprint Planning Meeting (whole team)**:
  + Refine prototype based on feedback.
  + Complete functional implementation and finalize testing.
* **Developers:**
  + Finish implementation of all functional requirements (goals, achievements, data analysis, trends visualization).
* **Test Lead:**
  + Conduct extensive testing, ensure coverage of all functional and non-functional requirements.
  + Document test outcomes clearly, ensuring verifiable results.
* **Designers:**
  + Update design documentation to match the final implementation.
* **Sprint Review & Retrospective Meeting**:
  + Demonstrate completed software.
  + Gather final feedback from tutors.

**Phase 5: Documentation and Final Report Compilation (Weeks 9–10: 10th–24th May)**

* **Whole Team:**
  + Assemble report sections collaboratively, adhering to the suggested page budget (20 pages total):
    - **Introduction (2 pages)**: Product Owner, Researchers
    - **Agile Process (2 pages)**: Scrum Master
    - **Requirements Specification (5 pages)**: Product Owner, Researchers
    - **Design (5 pages)**: Designers
    - **Software Testing (2 pages)**: Test Lead
    - **Reflection & Conclusion (4 pages)**: Whole team contributes based on personal reflections.
  + Include appendices (meeting minutes, GCF form, interview transcripts, test logs).

**Phase 6: Submission (Before 20:00 BST, Friday 2nd May 2025)**

* **Product Owner / Scrum Master:**
  + Ensure proofreading and coherence of the entire document.
  + Upload the final PDF document via Moodle.

**Weekly Scrum Meeting Guidelines (whole team, weekly):**

* Each member:
  + Updates on tasks completed and tasks remaining.
  + Discusses challenges encountered and solutions needed.
* Scrum Master:
  + Ensures productive meetings, keeps track of commitments, updates backlog, and logs minutes.

**Recommended Tools & Practices:**

* **Task Management:** Trello, Jira, or GitHub Projects
* **Version Control:** GitHub
* **Communication:** Slack, Discord
* **Documentation:** Google Docs, Overleaf (for collaborative LaTeX editing)
* **Testing:** JUnit (Java) or PyTest (Python), depending on your implementation choice.

By clearly distributing responsibilities, adhering strictly to Agile Scrum practices, and maintaining regular, documented communication, your group can effectively manage workload, remain on schedule, and deliver high-quality results.