**Requirement Analysis Document**

**Project Name**: Edu Quest: Coding Edition  
**Purpose**: To develop an interactive educational platform that teaches programming fundamentals to beginners, focusing on engagement and accessibility.

**1. Goals and Objectives**

* **Primary Goal**: Create a user-friendly software platform for students to learn coding basics through interactive lessons and gamified challenges.
* **Sprint Goals**:
  + Define and prioritize core features for the MVP (Minimum Viable Product).
  + Develop and test one complete module (e.g., loops).
  + Ensure the system supports basic user account management.

**2. Stakeholder Needs**

**Stakeholders Identified:**

* **Primary Users**:
  + **Students**: Learn coding interactively.
  + **Teachers**: Monitor and guide student progress.
* **Secondary Users**:
  + Parents: Track their child’s progress.
  + School Administrators: Integrate the platform into curriculums.

**Key Stakeholder Needs:**

* **Students**:
  + Clear, engaging lessons with visual and interactive components.
  + Gamified challenges to reinforce learning.
* **Teachers**:
  + Ability to create and manage classrooms.
  + Access to student progress reports.
* **Parents**:
  + Simple dashboards to view progress.

**3. Functional Requirements**

1. **User Management**:
   * Create, update, and delete student/teacher profiles.
   * Role-based access (students vs. teachers).
2. **Interactive Lessons**:
   * Modules on coding basics (e.g., loops, conditionals, variables).
   * Quizzes and challenges at the end of each lesson.
3. **Progress Tracking**:
   * Dashboard for students and teachers.
   * Visual representation of progress (e.g., progress bars, badges).
4. **Gamification**:
   * Reward systems (e.g., points, badges).
   * Leaderboards for friendly competition.

**4. Non-Functional Requirements**

* **Performance**:
  + Load times under 2 seconds for lessons and quizzes.
* **Scalability**:
  + Support for up to 100 concurrent users during MVP.
* **Security**:
  + Encrypted user data storage.
* **Usability**:
  + Intuitive navigation for users aged 10 and above.

**5. Constraints**

* **Time**: Deliver an MVP within the sprint duration (e.g., 2-4 weeks).
* **Budget**: Limited resources for tools and testing.
* **Technology**: Use open-source frameworks like React (front end) and Node.js (back end).

**6. Assumptions**

* Students and teachers will have access to the internet.
* The software will be deployed on web browsers and mobile devices.
* Teachers will provide initial guidance to students.

**7. User Stories (For Sprint and Trello Planning)**

**Epic 1: User Management**

* As a student, I want to create a profile so I can save my progress.
* As a teacher, I want to set up a classroom so I can monitor my students.

**Epic 2: Interactive Lessons**

* As a student, I want to complete lessons on loops so I can understand the basics of coding.
* As a teacher, I want to assign lessons to my class so they can learn systematically.

**Epic 3: Progress Tracking**

* As a student, I want to see my progress visually, so I know what to focus on next.
* As a teacher, I want to track the performance of my students to identify who needs extra help.

**8. Deliverables for the Sprint**

1. **Requirements Documentation**: Complete this analysis and update Trello with all user stories.
2. **Design Documentation**: Create wireframes for the user interface (UI).
3. **Working Software**: Develop a functional module (e.g., loops) for the MVP.
4. **Test Plans**: Write test cases for user management and lesson modules.
5. **Documentation**: Draft user guides for students and teachers.

**9. Trello Board Setup**

* **Columns**:
  + **Backlog**: Add all user stories and tasks.
  + **To Do**: Select sprint tasks (e.g., "Design lesson module UI").
  + **In Progress**: Tasks currently being worked on.
  + **Testing**: Completed tasks awaiting review.
  + **Done**: Completed and approved tasks.

**10. Acceptance Criteria**

* Students can create accounts and complete the "loops" lesson without errors.
* Teachers can create classrooms and assign lessons.
* Progress tracking shows accurate data for at least one student.