### User Stories

Sizing: is done using **Story Points**, which are units of measure for expressing the relative effort required to implement a user story. This estimate includes the complexity of the work, the risks or uncertainty involved, and the total volume of work. We'll use a modified Fibonacci sequence (1, 2, 3, 5, 8, 13) where a higher number means more effort.

**Baseline Story:** *As a Trainee, I want to view a catalog of all available courses* we'll assign it **2 points**. All other stories will be sized relative to this one.

#### 1. Must Have (MVP)

* **Admin: Create new user account** - **3 Points** (Involves a form, validation, and database write).
* **Admin: Assign/change user's role** - **3 Points** (Involves selecting a user and a role, then updating the database).
* **Trainee: View course catalog** - **2 Points** (Baseline - a straightforward data read and display).
* **Trainee: Register for a course** - **3 Points** (Requires linking a user to a course, a simple transaction).
* **Trainer: Create a new class** - **3 Points** (Requires a form and creating a new record in the database).
* **Trainer: Specify schedule and capacity for a class** - **3 Points** (Adds complexity to the class creation/edit form and data model).
* **Trainer: Open a class for booking** - **2 Points** (Likely changing a status flag and updating the UI).
* **Trainer: View a list of all my classes** - **2 Points** (A simple data read and display, similar to the course catalog).

**Total MVP Points: 21**

#### 2. Should Have

* **Admin: View list of all users** - **2 Points** (Similar to viewing courses, but for users).
* **Admin: Edit a user's details** - **3 Points** (Fetching user data, populating a form, and handling updates).
* **Admin: Delete a user** - **3 Points** (Requires confirmation logic and handling of user-related data, e.g., soft delete).
* **Trainee: View schedule for an enrolled course** - **2 Points** (Filtered data read and display).
* **Trainee: Book a specific class** - **5 Points** (A key transaction that must check capacity, create a booking record, and handle potential conflicts).
* **Trainee: See a list of my enrolled courses** - **2 Points** (A simple filtered view for the logged-in user).
* **Trainer: Close a class** - **2 Points** (Similar to opening a class, likely a status change).

**Total Should Have Points: 19**

#### 3. Could Have

* **Trainee: Pay for a course** - **13 Points** (This is a very large story, often called an "epic." It involves high complexity and risk due to integration with a third-party payment gateway, handling transactions, and ensuring security).
* **Admin: Generate basic reports** - **8 Points** (Reporting involves data aggregation and can have hidden complexity in querying and presenting the data, even for "basic" reports).

**Total Could Have Points: 21**

### Definition of Done (DoD) ✅

The **Definition of Done** is a universal checklist that must be completed for any user story to be considered "done." This ensures quality, consistency, and a shared understanding across the team.

A user story is considered **Done** only when all of the following criteria are met:

* **Code Complete & Reviewed**
  + All code has been written to implement the story's functionality.
  + Code has been peer-reviewed by at least one other developer.
  + Code adheres to the project's established coding standards and style guides.
  + The code has been successfully merged into the main development branch without conflicts.
* **Testing & Quality Assurance**
  + Unit tests have been written and are passing with sufficient code coverage.
  + Integration tests related to the feature have been written and are passing.
  + The feature has been manually tested by the QA team (or a designated tester) against all acceptance criteria.
  + No critical or high-severity bugs related to the story exist.
  + The feature passes all relevant regression tests.
* **Functionality & Acceptance**
  + The feature meets all acceptance criteria listed in the user story.
  + The functionality has been demonstrated to the Product Owner (PO).
  + The Product Owner has formally accepted the story.
* **Documentation & Release**
  + Any necessary technical documentation (e.g., API changes, architecture diagrams) has been updated.
  + The feature is deployed to a staging or User Acceptance Testing (UAT) environment, ready for release.