|  |  |
| --- | --- |
| Project name | Competetive Coding Arena (CCA) |

|  |  |  |
| --- | --- | --- |
| Project author | | |
| № | Names of Participants | FN |
| 1 | Ilker Raimov | 0MI0600050 |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The Competitive Coding Arena (CCA) is an online platform that enables students to compete in real-time coding competitions which consist of multiple programming tasks. CCA allows organizers to create coding challenges, define test cases, and manage users, while students can compete by writing C++ code to solve each task.  The platform will be developed using Golang with Chi or Gorilla Mux for routing, and will utilize Gorilla WebSocket for real-time communication. GORM will manage MySQL database interactions. (Not sure for the storage solution, might use Redis or any other instead) The front end will be built using the Golang html/template package with jQuery for asynchronous updates or some frontend framework.  Key features include:   * **Real-Time Code Execution**: Student code submissions (in C++) are run in Docker containers and tested against pre-defined cases in real-time. * **Competition Management**: Organizers can schedule and manage competitions and monitor students' progress live * **WebSocket Notifications**: Students and organizers will receive live updates on solution submissions, scores, and competition rankings.   The main user roles (actors in UML) are:   * **Anonymous User** – Can view information about the platform, upcoming competitions, and sample challenges. * **Student (Registered User)** – Can participate in coding competitions by submitting solutions to tasks, which are scored and displayed on a leaderboard. * **Organizer (Registered User)** – Can create competitions, manage users, define coding challenges, and monitor results. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse Information** | Users can view platform information, upcoming competitions, and sample coding tasks. | All users |
| * 1. **Register** | *Anonymous users can register, creating a student account.* | *Anonymous User* |
| * 1. **Create Competition** | *Organizers can create a new competition, define start/end times, and add coding tasks and tests.* | *Organizer* |
| * 1. **Add/Edit Coding Task** | *Organizers define/edit coding tasks, including problem descriptions, test cases, and scoring rules.* | *Organizer* |
| * 1. **Join Competition** | *Registered students can join active competitions, accessing tasks within them.* | *Student* |
| * 1. **Submit Solution** | *Students can submit C++ code solutions to competition tasks, which are executed and scored in real-time.* | *Student* |
| * 1. **Monitor Competition Progress** | *Organizers view live status updates, student progress, and scores for ongoing competitions.* | *Organizer* |
| * 1. **View Leaderboard** | *All registred users can view a leaderboard with rankings for the current competition.* | *Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Welcome** | Displays introductory information about CCA, upcoming competitions, and how to participate. | / |
| * 1. **Competitions** | Shows available competitions and past events, allowing browsing, filtering, and joining as applicable. | */competitions* |
| * 1. **Competition Dashboard** | Provides updates on competition progress, leaderboard, and scores. Available to active participants. | */competitions/{id}/dashboard* |
| * 1. **User Registration** | Allows Anonymous Users to register for a Student/Organizer account. | */register* |
| * 1. **Login** | Allows users to log into the CCA system. | */login* |
| * 1. **Create Competition** | Allows Organizers to create new competitions and manage existing ones. | */competitions/create* |
| * 1. **Task Editor** | Allows Organizers to define and edit coding tasks within a competition, including test cases and scores. | */competition/{id}/edit* |
| * 1. **Leaderboard** | Displays the leaderboard for a competition. | */competition/{id}/leaderboard* |
| * 1. **Dashboard** | Displays information and functionality depending on the role | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET user data for all users, POST new user data. | */api/users* |
| * 1. **User** | GET, PUT, DELETE user data for a specified user ID, according to role-based access restrictions. | */api/users/{userId}* |
| * 1. **Login** | POST user credentials to receive a security token for authenticated access. | */api/auth/login* |
| * 1. **Competitions** | GET list of competitions and POST a new competition | */api/competitions* |
| * 1. **Competition** | GET, PUT, DELETE competition data for a specific competition ID, with role-based restrictions. | */api/ competitions/{id}* |
| * 1. **Tasks** | GET list of coding tasks for a specified competition ID. | */api/competitions/{id}/tasks* |
| * 1. **Task** | GET, PUT, DELETE a specific coding task for a competition, including test cases and scoring data. | */api/competitions/{competitionId}/tasks/{taskId}* |
| * 1. **Submit Solution** | POST student C++ code submission to be compiled, tested, and scored in real-time. | */api/competitions/{competitionId}/tasks/{taskId}/submit* |
| * 1. **Leaderboard** | GET leaderboard information for an ongoing competition. | */api/competitions/{competitionId}/leaderboard* |