Front-end FDUC

|  |  |
| --- | --- |
| Use Case UC-1 & UC-2: | Search & View |
| Related Requirements: | REQ1 – REQ2 |
| Initiating Actor: | User |
| Actor’s Goal: | Search the programming challenges |
| Participating Actors: | Database |
| Preconditions: | User has stable Internet connection and updated browser |
| Success End Condition: | User is able to search and view desired programming challenge |
| Failed End Condition: | Search return empty list |
| Extension Points: |  |
| Flow of Events for Main Success Scenario: |  |
|  | **→** User opens web application through a web browser. |
|  | **←** Website loads a home page which has a trending challenges and an option to search challenges |
|  | → User clicks on the search box and enter a search query and press enter or click search button |
|  | Now UI will run a search query on the database. |
|  | ← UI will display the search result in a table format. Each challenge is a clickable link with metadata shown. |
|  | The data table will have options to filter and sort the entries. |
|  | → User will enter desired filter query into filter box. |
|  | → User will click on chosen challenge to view it. |
|  | ← Website will open the challenge page with detailed description, hint and optional notes tab for the particular challenge |
|  | Notes tab will be available for registered and logged in user. |

|  |  |
| --- | --- |
| Use Case UC-3 & UC-5: | Register & Mark |
| Related Requirements: | REQ-3 & REQ-4 |
| Initiating Actor: | User |
| Actor’s Goal: | A user registers an account in the system and store challenge specific information. |
| Participating Actors: | Database |
| Preconditions: | Unregistered user has stable Internet connection and updated browser |
| Success End Condition: | User is able to register a new account. User should be able to search, view & save desired programming challenge. |
| Failed End Condition: | User registration failed |
| Extension Points: |  |
| Flow of Events for Main Success Scenario: |  |
|  | **→** User opens web application through a web browser. |
|  | **←** Website loads a home page which has a trending challenges and an option to search challenges |
|  | → User click on register button |
|  | ← System loads user registration page. |
|  | → User inputs username, Full name and password. Then click register button. |
|  | Now system will query database in a backend to create username entry in users table. |
|  | ← Upon successful entry creation, website will display success message and keep the user logged in. |
|  | → User clicks on the search box and enter a search query and press enter or click search button |
|  | Now UI will run a search query on the database. |
|  | ← UI will display the search result in a table format. Each challenge is a clickable link with metadata shown. |
|  | → User will click on chosen challenge to view it. |
|  | ← Website will open the challenge page with detailed description, hint and optional notes tab for the particular challenge |
|  | Notes tab will be available for registered and logged in user. |
|  | → Now user will click on favorite icon (heart icon) to save the challenge in user profile. |
|  | UI will query the database to save this challenge in user profile table. |
|  | → UI will display the challenge display page with up to date user profile info. |

|  |  |
| --- | --- |
| Use Case UC-4: | Login |
| Related Requirements: | REQ5 |
| Initiating Actor: | User |
| Actor’s Goal: | Allow user to login and view saved challenges |
| Participating Actors: | Database |
| Preconditions: | Registered user has stable Internet connection and updated browser |
| Success End Condition: | User is able to login and view user profile |
| Failed End Condition: | User failed to login |
| Extension Points: |  |
| Flow of Events for Main Success Scenario: |  |
|  | **→** User opens web application through a web browser. |
|  | **←** Website loads a home page which has a trending challenges and an option to search challenges |
|  | → User click on login button |
|  | ← System loads user login page. |
|  | → User inputs username and password. Then click login button. |
|  | Now system will query database in a backend to validate user credentials in user table. |
|  | ← Upon successful user validation, website will load user profile page with user specific information like saved and solved challenges. |
|  | → User clicks on a favorite tab to view all saved challenges. |
|  | ← UI will display the favorite challenges list in the table format. |
|  | → User will click on chosen challenge to view it. |
|  | ← Website will open the challenge page with detailed description, hint and optional notes tab for the particular challenge |
|  | Notes tab will be available for registered and logged in user. |

back-end FDUCs

|  |  |
| --- | --- |
| Use Case UC-7: | Auto-Update |
| Related Requirements: | REQ 1 & 2 |
| Initiating Actor: | System Cron Job |
| Actor’s Goal: | System is able to fetch challenges from participating websites. |
| Participating Actors: | Participating challenge websites & Database |
| Preconditions: | Cron job is setup. |
| Success End Condition: | Database is updated with latest challenges from participating websites. |
| Failed End Condition: | Database entry failed to update. |
| Extension Points: |  |
| Flow of Events for Main Success Scenario: |  |
|  | **→** System starts the backend web scrapper scripts periodically as specified in cron job. |
|  | **←** Website will respond to web scrapper queries |
|  | → Web scrapper script will reformat the responses for local databases. |
|  | → Web scrapper script will save the latest result into the challenges table |

|  |  |
| --- | --- |
| Use Case UC-8: | Manual-Update |
| Related Requirements: | REQ 3 & 4 |
| Initiating Actor: | System Admin |
| Actor’s Goal: | System is able to fetch challenges from participating websites on demand. |
| Participating Actors: | Participating challenge websites & Database |
| Preconditions: | System user with admin privileges. |
| Success End Condition: | Database is updated with latest challenges from participating websites. |
| Failed End Condition: | Database entry failed to update. |
| Extension Points: |  |
| Flow of Events for Main Success Scenario: |  |
|  | → Admin user logs in to the system using a secure CLI session |
|  | → Admin user starts the backend web scrapper scripts using provided CLI. |
|  | **←** Website will respond to web scrapper queries |
|  | → Web scrapper script will reformat the responses for local databases. |
|  | → Web scrapper script will save the latest result into the challenges table |
|  | ← System will print a successful message on terminal. |