Back-end Developer Test

Scenario

Achievements

Our customers access their purchased courses via our Course Portal.

As part of this experience, users can unlock achievements:

Lessons Watched Achievements

- First Lesson Watched
- 5 Lessons Watched
- 10 Lessons Watched
- 25 Lessons Watched
- 50 Lessons Watched

Comments Written Achievements

- First Comment Written
- 3 Comments Written
- 5 Comments Written
- 10 Comments Written
- 20 Comments Written

Badges

Users also have a badge, which is determined by the number of achievements they have unlocked.

- Beginner: 0 Achievements
- Intermediate: 4 Achievements
- Advanced: 8 Achievements
- Master: 10 Achievements

Your Assignment

📝 **Unlocking Achievements**
You need to write the code that listens for user events and unlocks the relevant achievement.

For example:

- When a user writes a comment for the first time they unlock the “First Comment Written” achievement.
- When a user has already unlocked the “First Lesson Watched” achievement by watching a single video and then watches another four videos they unlock the “5 Lessons Watched” achievement.

AchievementUnlocked Event

When an achievement is unlocked an **AchievementUnlocked** event must be fired with a payload of:

achievement_name (string)

user (User Model)

BadgeUnlocked Event

When a user unlocks enough achievement to earn a new badge a **BadgeUnlocked** event must be fired with a payload of:

badge_name (string)

user (User Model)

Achievements Endpoint

There is an endpoint `users/{user}/achievements` that can be found in the `web` routes file, this must return the following:

unlocked_achievements (string[ ])

An array of the user’s unlocked achievements by name

next_available_achievements (string[ ])

An array of the next achievements the user can unlock by name.

Note: Only the next available achievement should be returned for each group of achievements.

👀 Example: If the user has unlocked the “5 Lessons Watched” and “First Comment Written” achievements only the “10 Lessons Watched” and “3 Comments Written“ achievements should be returned.

current_badge (string) The name of the user’s current badge.

next_badge (string) The name of the next badge the user can earn.

remaining_to_unlock_next_badge (int) The number of additional achievements the user must unlock to earn the next badge.

👀 Example: If a user has unlocked 5 achievements they must unlock an additional 3 achievements to earn the “Advanced” badge.

Test Coverage

You should write tests that cover all possible scenarios and would, in a real world project, make you confident there are no bugs and it is safe to deploy to production.

Laravel HTTP tests documentation can be found at the following url:

https://laravel.com/docs/10.x/http-tests

Additional Information

Installation

1. Download the repo from https://drive.google.com/file/d/1paeaIyPu5NSGptbpC-biddZ8aovS6KDo/view?usp=sharing
2. From the root directory run `composer install`
3. You must have a MySQL database running locally
4. Update the database details in `.env` to match your local setup
5. Run `php artisan migrate` to setup the database tables

The project is a standard Laravel 10 application, additional installation documentation can be found at https://laravel.com/docs/10.x/installation

User Events

The course portal fires events when a user carries out specific actions:

LessonWatched

Fired when a user watches a lesson.

CommentWritten

Fired when a user writes a comment.

These already exist in the codebase and can be found in `app/Events`.

Your task is not to implement the logic that fires these events, you can work on the assumption that another part of the system will be responsible for this.

Your assignment is to just listen for these events and ensure the correct achievements & badges are unlocked as outlined in the previous section of this document.

User Model

The following relationships are available on the user model:

watched

This will return an eloquent relationship for lessons watched by a user.

comments

This will return an eloquent relationship for comments written by a user.

How Your Submission Will Be Assessed

Your submission will be assessed on the following:

- **Meets All Criteria -** The solution meets all the criteria outlined in this document.
- **Code Quality** - The code is readable, elegant and could be easily expanded in future (for example, the addition of new achievements or badges)
- **Test Coverage -** Your tests cover all possible scenarios and we would feel confident there are no issues deploying your code to production.

Timeline: Expect to spend 4-6 hours working on the assignment.

Good luck!