Backend coding challenge

We will not ask you to design a fancy algorithm. In the end that is not what you are going to do on a daily basis if you work for us. Instead, please deliver something **production-ready** which makes you proud and follows the best engineering practices. Please focus on the requirements provided here and avoid over-engineering your solution. Focus on showing us how you would have solved a problem in your daily work.

We ask you to provide a backend solution, ideally involving:

- Java 8+
- Spring Boot 2.5+ (You may use Spring Initializer to bootstrap the project - https://start.spring.io/)
- Maven/Gradle
- Database of your choice (In-Memory database is acceptable)
- Docker or Docker Compose

Case study

George is a local travel guide who provides private hiking tours. Hikers can go on 3 different trails: **Shire**, **Gondor** and **Mordor**. George would like to have a booking system that allows hikers to book a hike.

Business Information

Trails

```
{
  "name": "Shire",
  "startAt": "07:00",
  "endAt": "09:00",
  "unitPrice": 29.90
}
```

```
{
  "name": "Gondor",
  "startAt": "10:00",
  "endAt": "13:00",
  "unitPrice": 59.90
}
```

```
{
  "name": "Mordor",
  "startAt": "14:00",
  "endAt": "19:00",
  "unitPrice": 99.90
}
```

`unitPrice` represents the price per hiker.

Conditions

- A booking can contain multiple hikers. A hiker should provide the following information: `firstName` , `lastName` and `age` .
- A booking can be done for one trail only.

Task

You should provide Restful APIs that perform the following tasks:

- View all trails
- View a selected trail
- Book a selected trail
- View a booking
- Cancel a booking

Assumptions

Given the limited amount of time to complete the task, you may assume the following:

- Any authentication/authorisation functionality is **NOT** required.

Technical Requirements

Your application should meet the following technical requirements:

- The trails' data should be initialized in the database on application start up.
- Have an example of a unit test (Additional unit tests are welcomed if there is spare time).
- Have an example of an integration test (Additional integration tests are welcomed if there is spare time).
- A README file with detailed instructions/commands to run it.
- It should run out of the box without any installations or configurations (databases, JDKs, etc) required by us.

Important

- Please deliver your solution within 7 days. You are free to spend any reasonable amount of time that you deem sufficient to achieve the technical requirements and makes you proud of your solution.
- Your solution will be compared against other candidates applying for the same job position, in terms of code quality and production-readiness.
- Please do not publish the solution to your public GitHub profile or something similar. You may send us your solution by sharing it as a zip file or via your private cloud account (Google Drive, OneDrive, etc).