Project with focus on backend

REST-interface, form architecture, validation

Preface

- There are more tasks than can possibly be solved within one day.
- Your approach to the solution, code structure and overall quality are more important than the completion of the tasks
- The layout is not important. But it should but work.
- The project has to run through the build process at the end. The call `mvn clean verify` needs to be successful.

Task

The web application can be started via `mvn spring-boot:run`. The start page can be opened with `http://localhost:8080/klt`. The page shows an application form with fields for first name and last name.

The application will grow step by step with these tasks:

1. It should be possible to submit the form. If all form elements are filled a page should appear saying: "Thank you for your application". If not all elements are filled the form should reappear with the empty fields being highlighted.
2. Implement a new age validation REST endpoint within the existing JAX-RS controller. When called `GET /klt/rest/age/{age}` (with {age} being a number) following status should be returned:
   - values < 18 status: "too young"
   - values > 67 status: "too old"
   - if the value is a prime status: "funny"
   - in all other cases status: "OK"
3. Write junit tests for this method in the REST-service proving the correctness of your implementation. The test has to run within the build process.
4. The application data from (1) has to be saved. The saved applications do not have to persist after a restart.
5. There should be a new REST endpoint returning all saved applications in a list.
6. There should be a new website showing all saved applications in a simple table.
7. A new field labelled "age" should be added to the application form in (1). This age information should also be saved for each application and be shown in (4) and (5).
8. Once the user leaves the age field, an instant AJAX/REST validation should run using the rest endpoint in (2) and the resulting status should be shown to the user.
9. Change the form validation process so that it is not possible to save the application if the status. is not "OK" or "funny".
10. Create a new endpoint in the rest service to delete a specific application.
11. All saved applications in the table of (5) should additionally get a delete button. When this button is pressed the specific application should be deleted using the rest endpoint in (9) and the application table should reappear without the just deleted.