**Java applications – Seminar – Homework**

You will find **help** for the task at the end of the document!

**Form groups of two and solve the task together in project work.** With this, you practice general project tasks at companies and the project use of Git. Describe in the documentation which of the two people did which part of the task.

**If someone cannot find a partner (e.g. they are odd in the group), do the GitHub task in project work using two GitHub accounts.**

In the <https://drive.google.com/drive/folders/1wxduPpMWGPkZ-arnHWX6-S1qCyy7R-d7?usp=sharing>

folder can find database sources, from which the group must choose one and use it in its task. **Only one group can choose the same database. The database can be expanded with new tables.**

In the Teams group, you will find the **Pair and Database selection for the Homework.xlsx** file. Enter your partner’s name and the selected database.

In the task, you have to create a **server-side web application** with the Java Spring Boot framework.

**Adapt the theme of your website to the chosen database.** Collect resources (images, texts, ...) from the Internet.

**Sum 20 points**

Create a website for a company, organization, foundation... that meets the following requirements:

1. On the first page, present the company on a spectacular website **(2 points)**
2. Have Registration, Login option **(3 points)**

- The "Login" menu item is visible if the user is not logged in.

- The "Logout" menu item is visible when the user is logged in.

- Display the logged-in user on the system header, if logged in

1. Distinguish at least 3 user roles: **(3 points)**

Admin, User, Visitor

The appearance of the menu items and the availability of the pages vary depending on

which user is using the site. (e.g. admin page)

1. Have a page where you display data from the chosen database **(2 points)**

Use the data of 3 tables from the database.

1. One page should have a contact form, which you can use to send a message **(2 points)**

and can be sent to the site owner. Check the form correct filling with

server-side validation. Save the sent form data to the database.

1. Make it possible to view previous point messages on a table from the database. **(2 points)**

in reverse chronological order. Print the sending time and the sender name

for each message. In the case of a non-logged-in user: "Guest".

1. Implement a RESTful API in the application. **(3 points)**

Test API functionality with both cURL and Postman. Take screenshots

of the tests to the documentation

1. Use the GitHub (github.com) version control system. **(2 points)**

**(Mandatory element! The source will be checked based on it)**

1. Use the project work method on GitHub: **(2 points)**

Display which part of the project was prepared by which group member.

Don't just upload the finished application in one step, but also the partial states

in at least 5 steps.

**Choose your own name on the GIT, so for every task can be identified who uploaded that.**

Create a **run.jar** package file for your application. Place the package file in the root folder of your Github project. Save (Export) your database as **data.sql**, from which **the database can be imported in one step** into the XAMPP database system.

Your application's **application.properties** file should contain the following settings:

spring.datasource.url=jdbc:mysql://${MYSQL\_HOST:localhost}:3306/**exercise**

spring.datasource.username=**root**

spring.datasource.password=

spring.datasource.driver-class-name =com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

**The application started with the JAR file must work with the imported database!**

**Create at least 15 pages of documentation (in PDF format)** describing how the app works with screenshots. **(Mandatory element!)** In the documentation, give the URL address of your GitHub project and **describe in detail how and where you implemented the previous tasks in the application**. It is important that you describe it in the documentation, because the task will be checked based on this.

**Help for the parts of the application**

|  |  |
| --- | --- |
| **Tasks** | **Chapter in the Seminar PDF file** |
| Spectacular website | with the help of what you learned in Web programming-1 |
| Creating JAR file | Creating an executable JAR file with Maven |
| Registration, Login option | Spring-Boot – Security |
| user roles | Spring-Boot – Security |
| Form, Server-side validation | Spring Forms |
| Database write/read | Spring-Boot-Database-JPA |
| GitHub usage | GitHub-GitLab usage.docx |
| RESTful API | Spring-Boot – RESTful API |