Today I am going to tell you about my test java project. The project name is Music notebook.

Let's take a look at the first slide.
This slide represents goals of the project. The main goal is to make a project with the following structure. The project should be divided into two different runnable applications. As you may see it is the Web-app that provides UI to customers and the Rest-app that provides a rest-api for database access. To receive data, the Web-app requests it from t4he Rest-app. Also required that this project be built with Spring framework.

Ok, let's go to the next slide.
Here you can see my project theme. Music Notebook is an application for recording descriptions of your favorite songs and making playlists from them. Music notebook must provide:

- CRUD for songs
- CRUD for playlists
- Sort songs by playlists (That means we should be able to add or delete songs to playlist, and when we delete this playlist its songs are not deleted)
- Filter song by release date (We can specify the start and end dates of the period when songs were released and then and then only these songs will be shown)

Ok now we will take a look at the database structure
Songs and playlists are tables for the main entities used in the application. And there is also an additional table for implementing many-to-many relationships.I chose this structure because a playlist can contain many songs, and the same song can be included in several playlists at once.

Ok, Let's go next.
This slide represents the project structure. It was a multi module maven project and here you can see all modules which were created. Let's see each of them.
Model provides root instances of songs and playlists which will be used by other modules.li
Test database provides test database for all implementations of dao-api.
Service-api and dao-api are API layer.
Dao-jdbc is JDBC implementation dao-api.
Service-dao and Service-rest modules implement Service-api.
Service-dao is the main service for Rest-app and it can use dao-api implementations to get database access.
Service-rest is the main service for Web-app and this service can send requests using Rest-API and process responses.

And on top, there are two main modules that represent the previously described applications.

Now I'm going to look at these two modules in more detail in the next slides.
Here you may see the technologies list.
The first four are the main modules of the spring framework that were used in this application.This is a logging tool. Logs are output to the console and saved to a file. Jupiter, Mockito and MockMVC are tools for testing. For rest-app, implemented independent testing of each layer. Jackson provides the ability to serialize and deserialize our objects. H2, MySQL and Postgres are available databases. Swagger for generating documentation. Hibernate validator provides an easy way to validate incoming data. Exception handling has been implemented with Controller Advice.

Let's move to the next slide
There are two additional technologies here. It is Bootstrap and Thymeleaf. Bootstrap is used to develop HTML pages and implement responsive design. Thymeleaf is a Java template engine. It fills HTML templates with data using java code.

Let's go next.
This slide shows how the design differs depending on the size of the device. Some columns stop showing and a collapse button appears.

Ok, Let's move to the last slide “Building and running the application”
Apache maven project builder is used.The Maven builds two executable jar files. When you run these files, the application is automatically deployed to the embedded Tomcat server. Added the ability to run an application inside a docker container. Can be run separately as standalone applications and as a single application with docker-compose (with mySQL database).

Well it was the last slide of my presentation. Thank you for your attention. If you have any questions I will be glad to answer.