# Workshop - MVC Project:

# Next Level Technologies

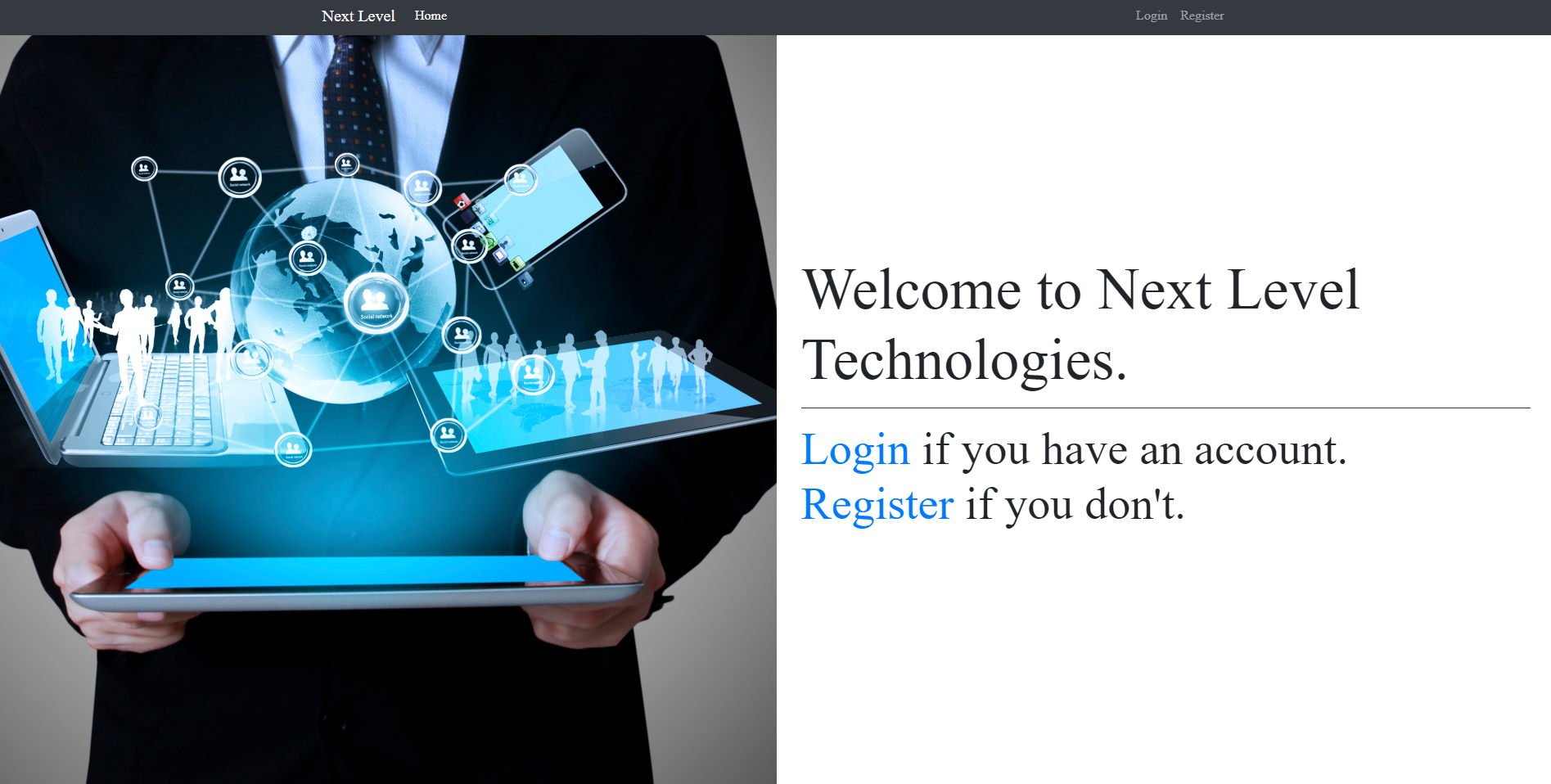
You have been employed by the Next Level Technologies Ltd. to finish the database layer, which supports basic functionality like importing data and exporting some results.

## Functionality Overview

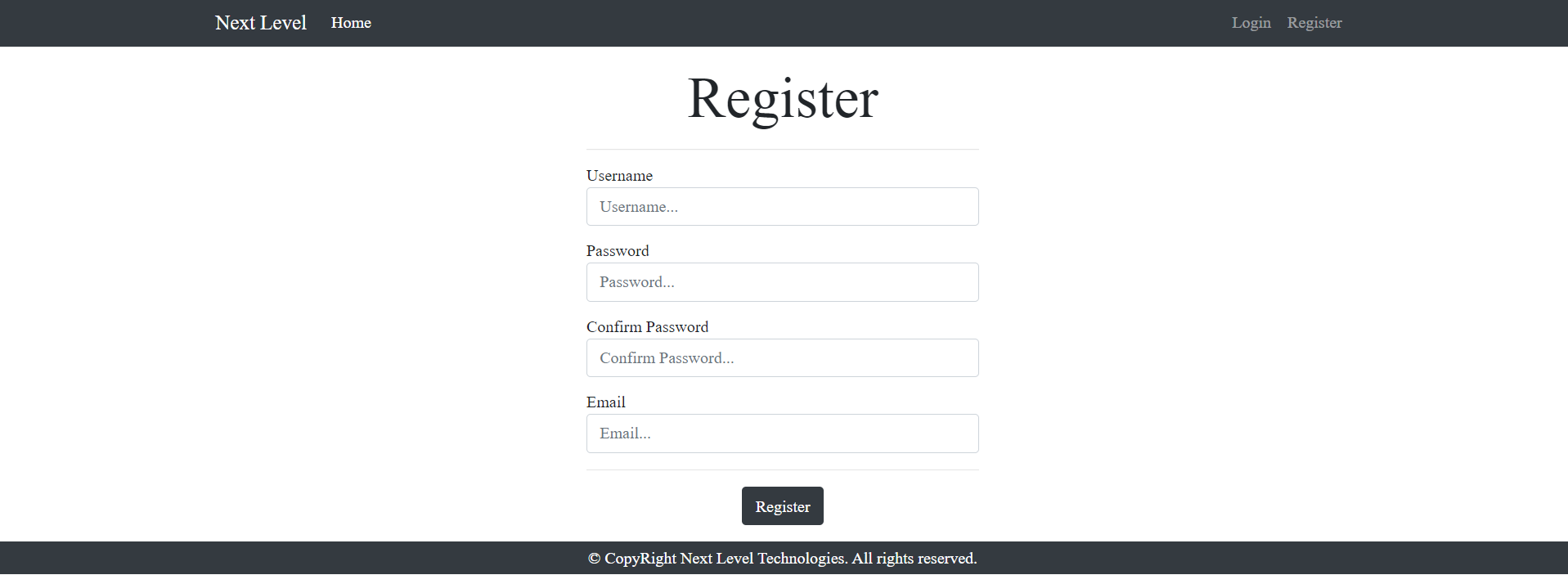
The firm has hired you as their application developer, to implement the **database layer**. The application should be able to easily **import** hard-formatted data and **support functionality** for also **exporting** the imported data. The application is called – **NLT**.

Look at the pictures below to see what must happen:

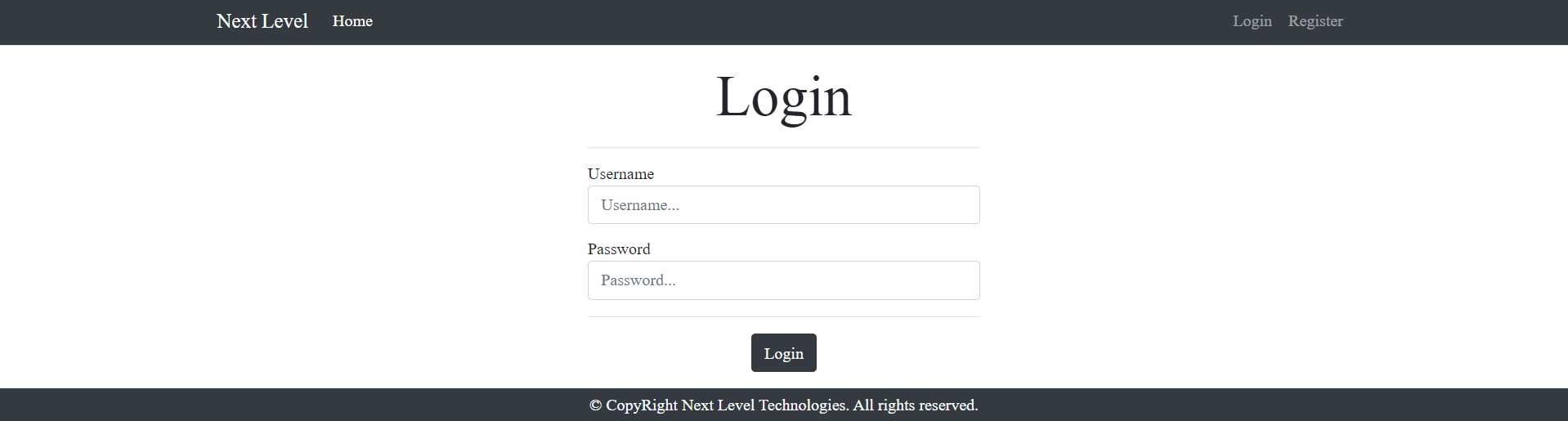
* Home page before logging in importing anything (route "/"):



* Register page (route "/users/register"):



* Login page (route "/users/login"):



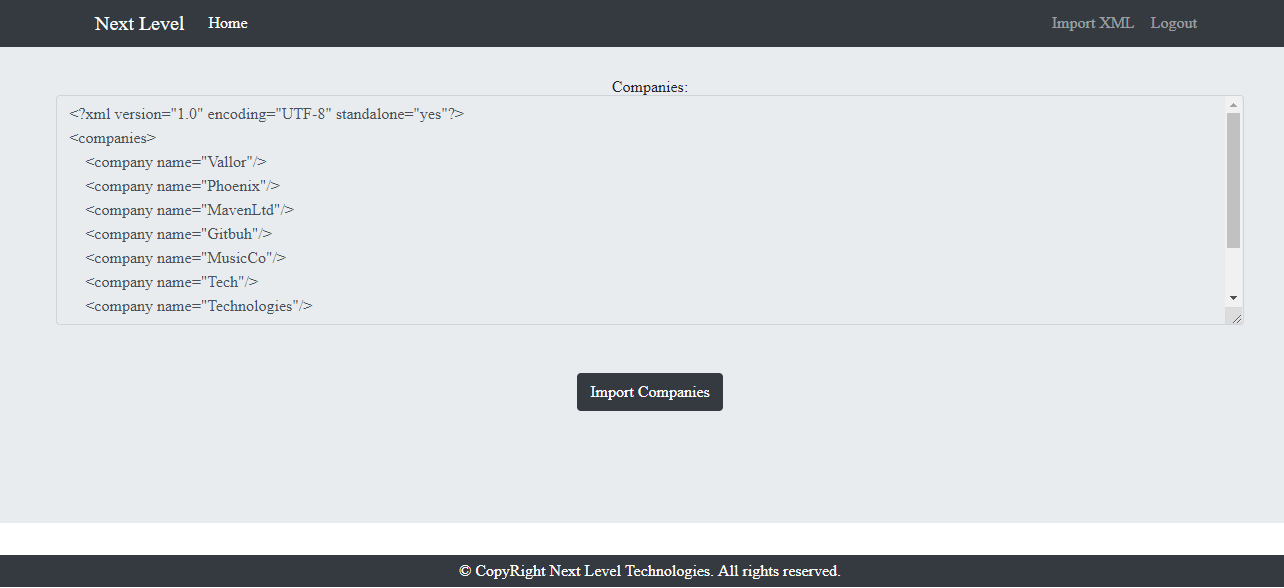
* Home page after login (route "/home"):



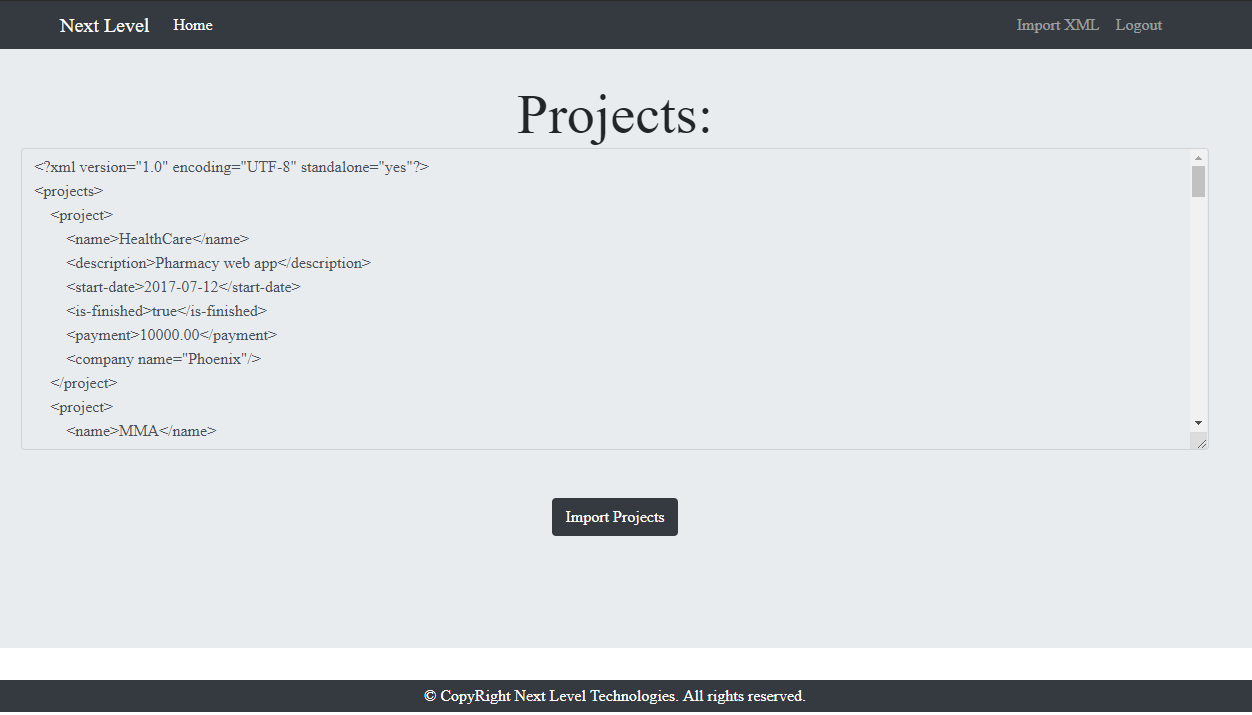
* Import XML page before importing anything (route = "/import"):



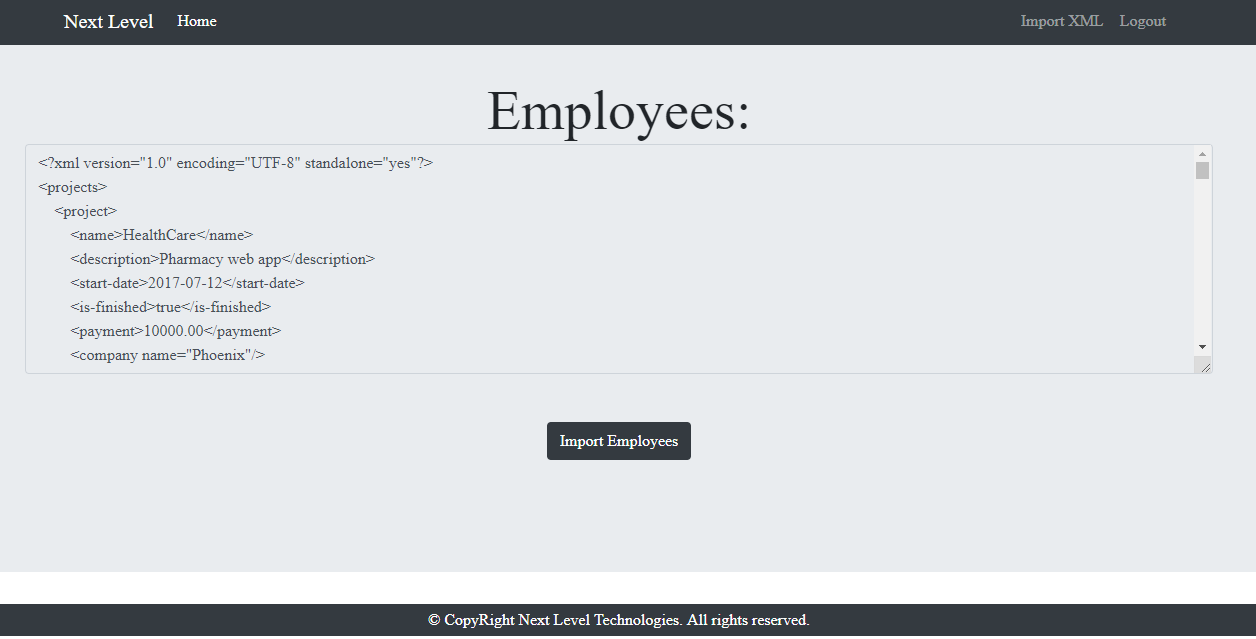
* Import Companies first (route = "/import/companies"):



* Import Projects second (route = "/import/projects"):



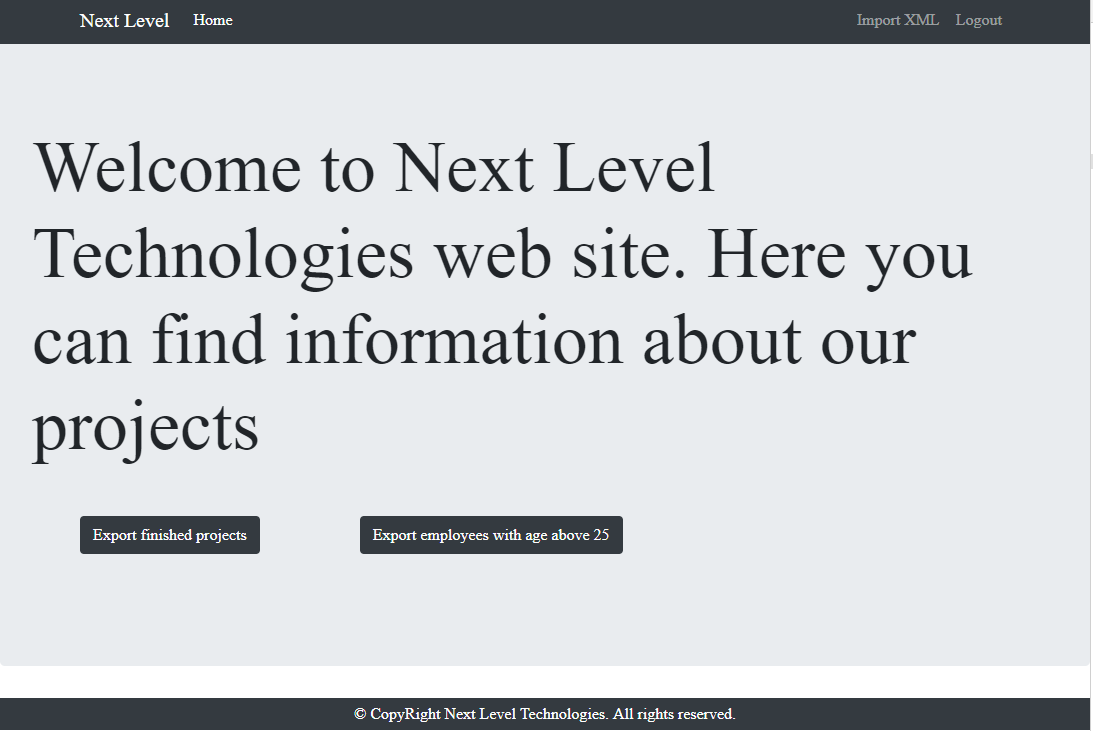
* Import Employees last (route = "/import/employees"):



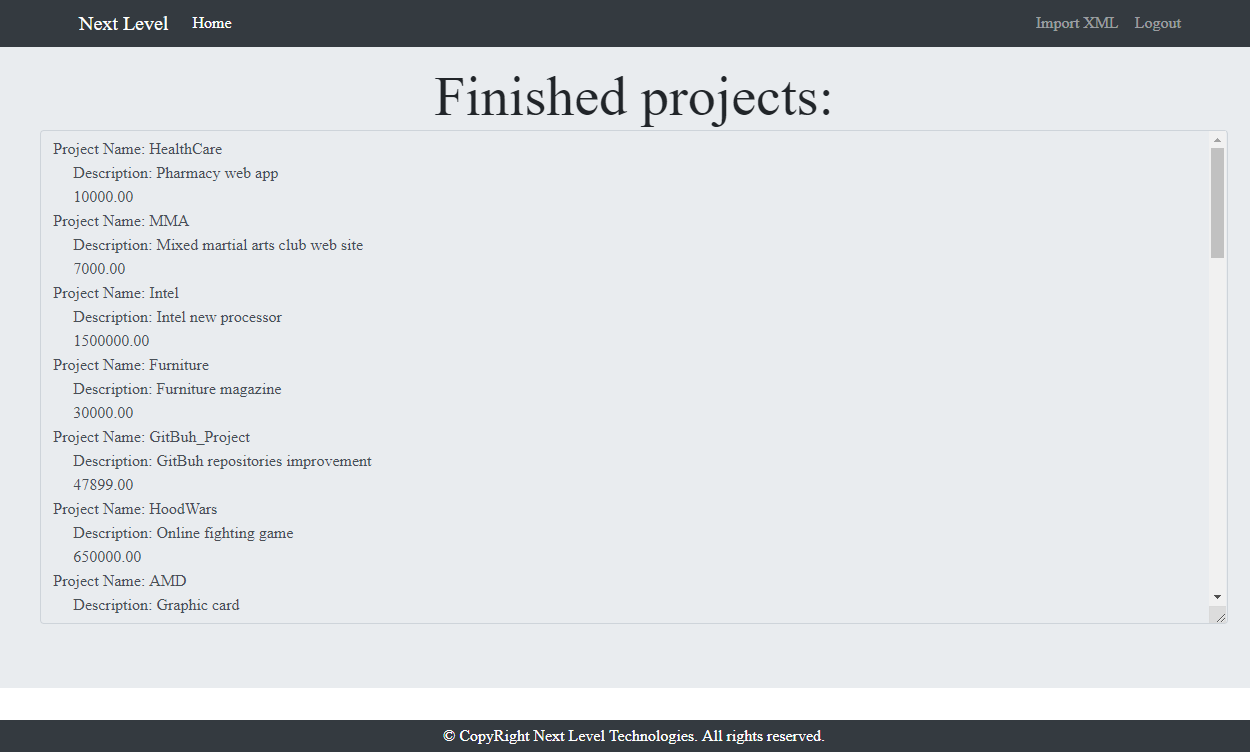
* Import XML page after importing the given data (route = "/import"):



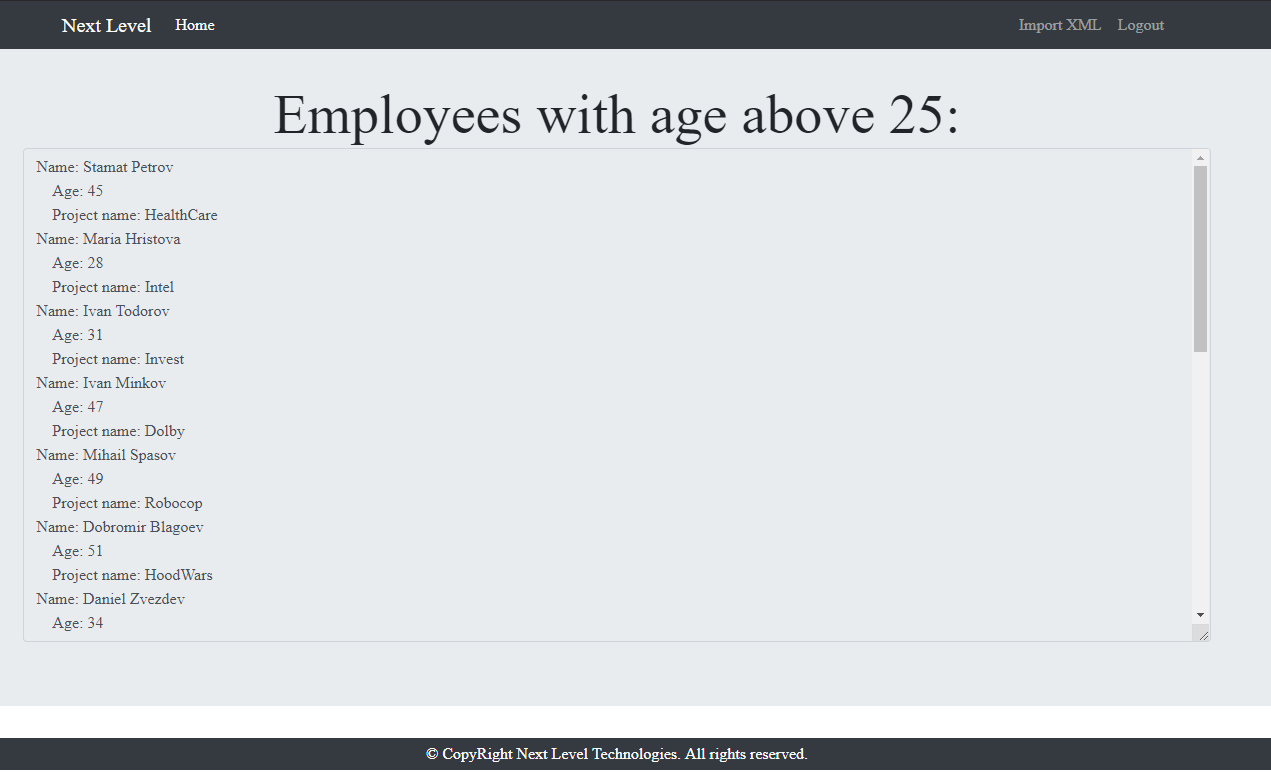
* Home page after successful imports (route ="/home"):



* Export Finished projects (route = "/export/projects-if-finished"):



* Export Employees with age above 25 (route = "/export/employees-above-25"):



## Project Skeleton Overview

You will be given a **Skeleton**, containing a **certain architecture(MVC)** with **several classes**, some of which – completely empty. The **Skeleton** will include the **files** with which you will **seed** the **database**.

## Model Definition

There are 3 main models that the **nlt** database application should contain in its functionality.

Design them in the **most appropriate** way, considering the following **data constraints**:

### Company

* id – **integer** number, **primary identification field**.
* name – a **string** (**required**).

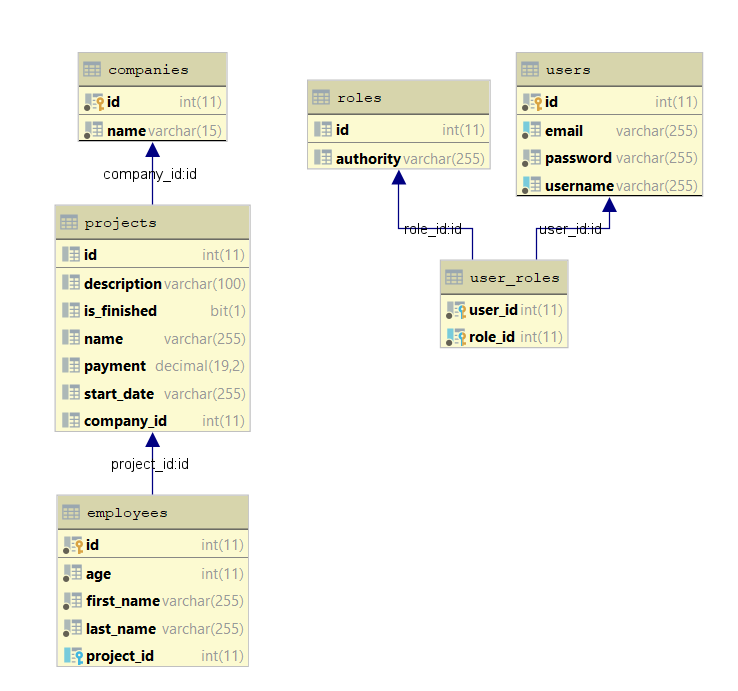
### Project

* id – **integer** number, **primary identification field**.
* name – a **string** (**required**).
* description – a **string** (**required**).
* is\_finished – a Boolean.
* payment – Big Decimal (**required**).
* start\_date – a String !
* company – a **Company** entity (**required**).

### Employee

* id – **integer** number, **primary identification field**.
* first\_name – a **string** (**required**).
* last\_name – a **string** (**required**).
* age – a Integer(**required**).
* project – a Project entity(**required**).

**NOTE**: Name the entities and their class members, **exactly** in the **format stated** above. Do not name them in snake case with the dashes, of course. But if a field is specified as first\_name, you are to name it firstName.



#### Relationships

The NLT decided to give you a little hint about the more complex relationships in the database, so that you can implement it correctly.

One Project may have only one Company, and one Company may have many Projects.

One Project may have many Employees, and one Employee may be appointed to only one Project.

## Data Import

Use the provided files to populate the database with data. Import all the information from those files into the database.

**You are not allowed to modify the provided files.**

**ANY INCORRECT** data should be **ignored** and a message appropriate should be printed.

* **NOTE**: An incorrect data input is an input which is **missing required fields**.

### XML Import

The **nlt** have prepared some XML data for you to import.

#### Company (companies.xml)

|  |
| --- |
| **compamies.xml** |
| *<?***xml version="1.0" encoding="UTF-8" standalone="yes"***?>* <**companies**>  <**company name="Vallor"**/>  <**company name="Phoenix"**/>  <**company name="MavenLtd"**/>  <**company name="Gitbuh"**/>  <**company name="MusicCo"**/>  <**company name="Tech"**/>  <**company name="Technologies"**/>  <**company name="FutureInvests"**/>  <**company name="DesignGroup"**/> </**companies**> |

#### Project (projects.xml)

|  |
| --- |
| **projects.xml** |
| *<?***xml version="1.0" encoding="UTF-8" standalone="yes"***?>* <**projects**>  <**project**>  <**name**>HealthCare</**name**>  <**description**>Pharmacy web app</**description**>  <**start-date**>2017-07-12</**start-date**>  <**is-finished**>true</**is-finished**>  <**payment**>10000.00</**payment**>  <**company name="Phoenix"**/>  </**project**>  <**project**>  <**name**>MMA</**name**>  <**description**>Mixed martial arts club web site</**description**>  <**start-date**>2018-09-21</**start-date**>  <**is-finished**>true</**is-finished**>  <**payment**>7000.00</**payment**>  <**company name="Vallor"**/>  </**project**>  ... </**projects**> |

#### Employee (employees.xml)

|  |
| --- |
| **employees.xml** |
| *<?***xml version="1.0" encoding="UTF-8" standalone="yes"***?>* <**employees**>  <**employee**>  <**first-name**>Stamat</**first-name**>  <**last-name**>Petrov</**last-name**>  <**age**>45</**age**>  <**project**>  <**name**>HealthCare</**name**>  <**description**>Pharmacy web app</**description**>  <**start-date**>2017-07-12</**start-date**>  <**is-finished**>true</**is-finished**>  <**payment**>10000.00</**payment**>  <**company name="Phoenix"**/>  </**project**>  </**employee**>  ...  </**employees**> |

## Data Export

Get ready to export the data you’ve imported in the previous task. Here you will have some pretty complex database querying. Export the data in the formats specified below.

#### Finished Projects

**Export all projects** which have **been finished**:

* Extract from the database, the project.

#### Employees with age above 25

**Export** the **employees** with age above 25.

* Export the **employee’s** **full** **name**, **age and project name.**