# Exam – SoftUni Booking

You need to create a **web application** where users **see and rent rooms**. Usually the users will visit the page, navigate either to **signing up or sign in**. When they are logged they will have the option to view **all offers** listed. They can either **add** a new offer, **view all their** offersor **edit** or **delete** existing, only if **they** added it.

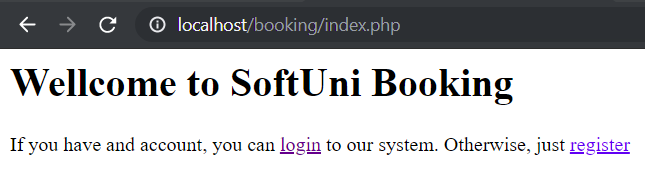
# 1. Expected Result

General rules:

1. Each **title** is **Heading 1**
2. All **attribute values and tags** are **lowercase**
3. All name fields are lower\_snake\_case variant of the text/value
4. All **tables** have **head** and if they are filled with data, they have a **body**
5. All **success messages** are stored in the **session** under the “success” key and are **immediately deleted** from session after visualization. They are visualized on the **next line after the title** in a **paragraph** with style color:green.
6. All **error messages** are visualized on the **next line after the title** in a **paragraph** with style color=red.

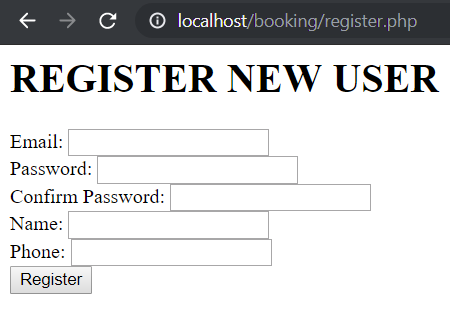
**index.php:**

* When user **is** **not** logged

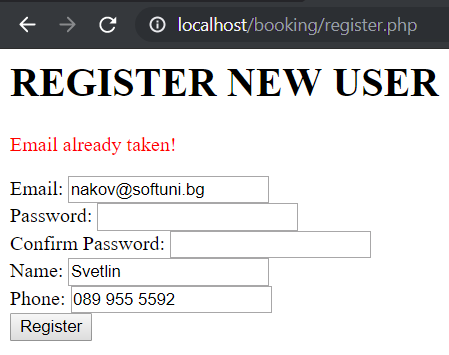


**register.php:**

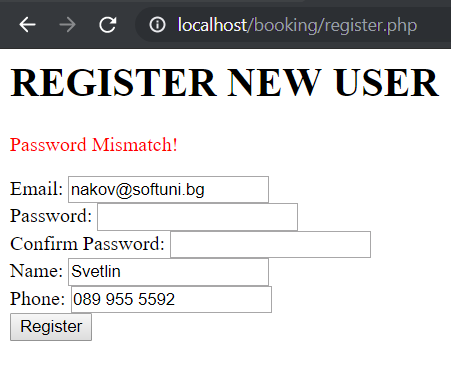
* **Before** form is submitted



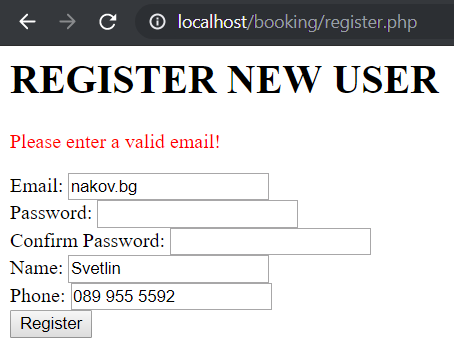
* Form **is submitted**, but there are **email** **duplicates**



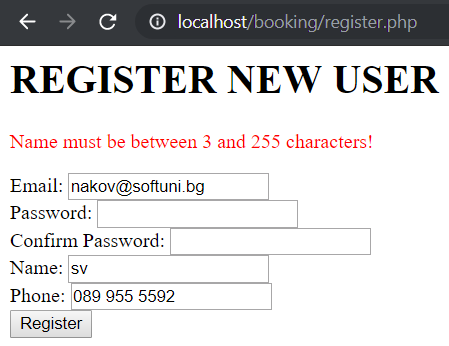
* Form **is submitted**, but the **password** and the **confirm password** field differs:



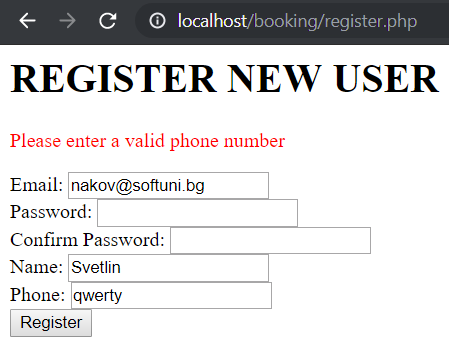
* Form **is submitted**, but **email** **validation** did not pass:



* Form is **submitted**, but **name** isn’t between MIN\_LENGHT and MAX\_LENGTH

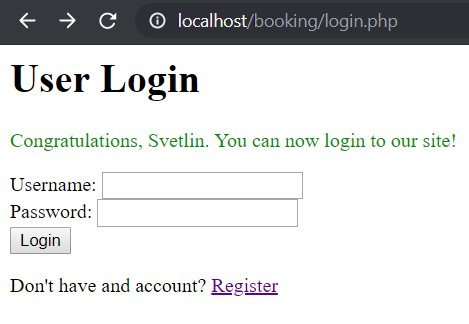


* Form is **submitted**, but **phone** **validation** did not pass:



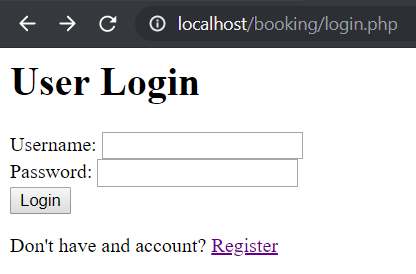
Show the **error only for the first field the validation did not pass** (they might all be out of range). The message is in the format:

* + For **text** fields: “{Field Name} must be between {MIN\_LENGTH} and {MAX\_LENGTH} characters!”.
  + For not valid **phone “Please enter a valid phone number”**
* **After** successful submission **“Congratulations, {Name}. You can now login to our site!”**

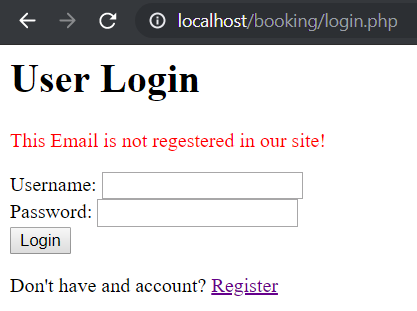


**login.php:**

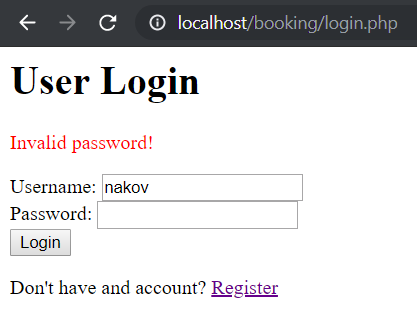
* **Before** form is submitted



* Form **is submitted**, but this **email** is not registered:

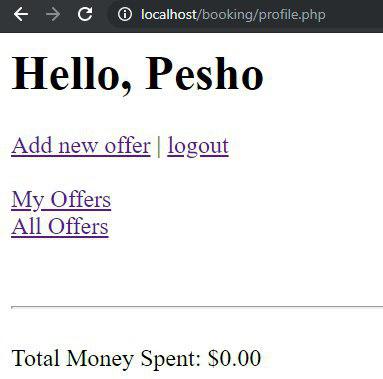


* Form **is submitted**, but the **password** is not the same:



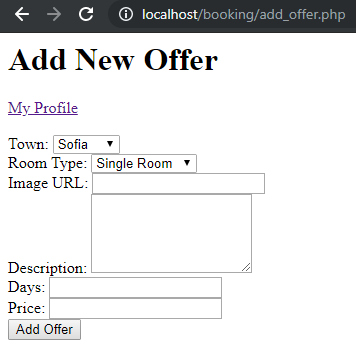
* **After** successful login:

You should get a greeting message with your **name.** You must have 4 links for adding new offer, for **logout** for viewing your offers and for viewing all offers, then <hr> and Total Money you spent with $ and formatted to the second decimal place



**add\_offer.php** should look like this:

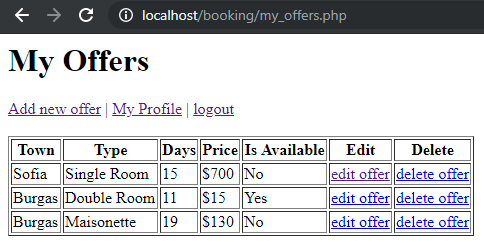
when adding default for **is\_occupied** is **false** and **added\_on** is **current\_timestamp**



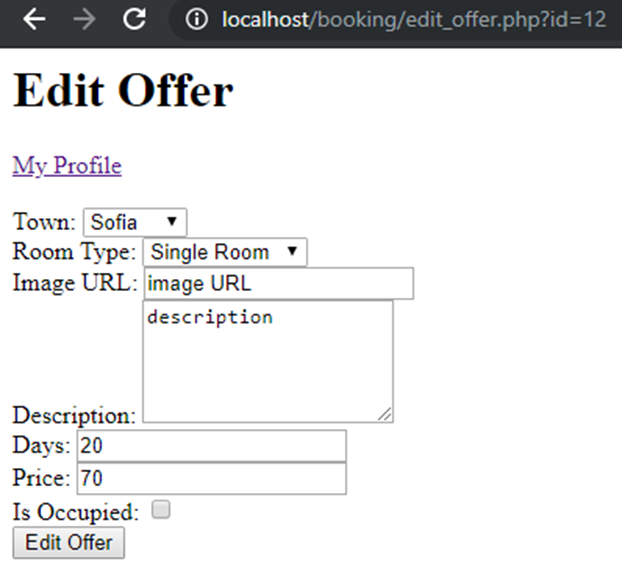
After adding an offer you should be reditected to **my\_offers.php**

my\_offers.php:

Sorted by Date(added\_on)Descending



**edit\_offer.php?id={offer\_id}** (Edit existing offer):



* After **successful submission**, the operation is **edited**, **redirect** to the list of my offers **my\_offers.php**

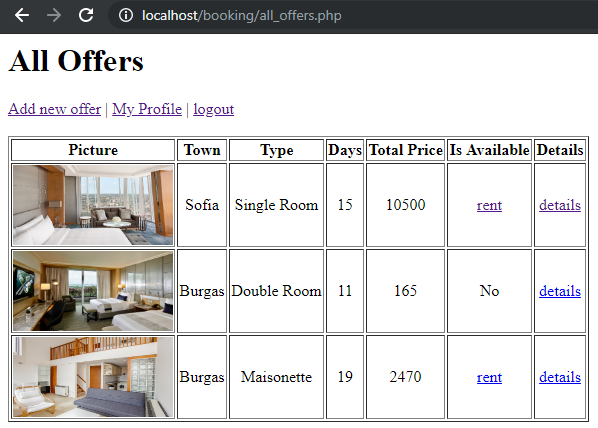
**delete\_offer.php?id={offer\_id}** (Delete existing offer):

* **Deletes** the offer and **redirects** to **my\_offers.php**.

**all\_offers.php:**

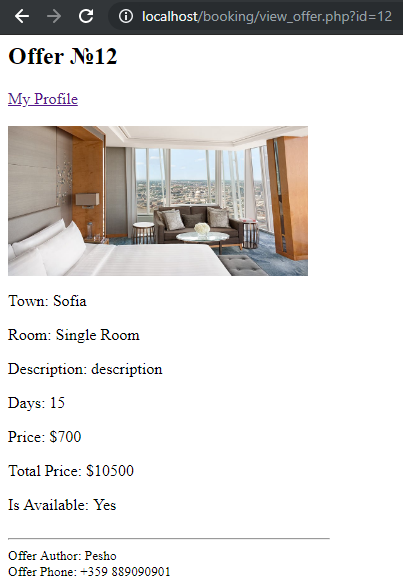
Sorted by Date(added\_on)Descending Total Price is (days\*price),

When you clicked to link – rent, is\_occupied should change his status to true, and user's money spent(money\_spent) should incresed by total price.

****

view\_offer.php?id={offer\_id} When you click “details” from all\_offers.php you should be able to see the offer **number (id)**,

offer **picture**, **town**, room **type**, **days**, **single** day price, **total** price, is it **available**, <hr> and then, <small> user’s **phone** and **name** about the offer you selected. This option is available only for logged in users.



# Database

Create a database “softuni\_booking” and add the following tables:

# Data Definition

# DDL – Users

|  |  |  |
| --- | --- | --- |
| **users** | | |
| **Column Name** | **Type** | **Constraints** |
| email | String | 4…255. Unique. |
| password | String | 4…255 |
| name | String | 4…255 |
| phone | String | 4…255 |
| money\_spent | Decimal(10,2) | 0...4294967295 |

# DDL – Towns

|  |  |  |
| --- | --- | --- |
| **Towns** | | |
| **Column Name** | **Type** | **Constraints** |
| name | String | 3…255 |

# DDL – Rooms

|  |  |  |
| --- | --- | --- |
| **Rooms** | | |
| **Column Name** | **Type** | **Constraints** |
| Type | String | 3…255 |

# DDL – Offers

|  |  |  |
| --- | --- | --- |
| **Offers** | | |
| **Column Name** | **Type** | **Constraints** |
| price | Decimal(10,2) | -10^38+1...-10^38-1 |
| days | int | 1...4294967295 |
| description | text | 10…10000 |
| picture\_url | text | 10…10000 |
| room\_id | int | 1...4294967295 |
| town\_id | int | 1...4294967295 |
| user\_id | int | 1...4294967295 |
| is\_occupied | bit | true/false (0/1) |
| added\_on | timestamp | current\_timestamp |

# DML – Add Towns

**Add** the following Towns

|  |
| --- |
| **name** |
| Sofia |
| Varna |
| Plovdiv |
| Burgas |
| Pleven |
| Other |

# DML – Add Rooms

**Add** the following Rooms: Save the DDL in export.sql file in the top-level directory of your project.

|  |
| --- |
| **Type** |
| Single Room |
| Double Room |
| Maisonette |

# HTML

# Register Form

**Create register form**. Follow the constraints below:

1. Email with minimum length of **4** symbols
2. Password with minimum length of **4** symbols
3. Confirm password with minimum length of **4** symbols
4. Name with minimum length of **5** symbols
5. **Phone** with minimum length of **6** symbols
6. Button, submitting the form with value “**Register**”

All fields are **required**. Use a convention for name attributes: **lower\_snake\_case**.

# Login Page

1. Username
2. Password
3. Button, submitting the form with text “**Login**”

All fields are **required**. Use a convention for name attributes: **lower\_snake\_case**.

# New Offer Page

1. Town which is **empty** **dropdown** (will be later populated with PHP)
2. Room Type which is **empty** **dropdown** (will be later populated with PHP)
3. Image string containing url
4. Description containing **10 to 10,000** symbols
5. Days integer with minimum length of **1** symbols and max length – **50**
6. Price decimal with minimum length of **1** symbols and max length – **50**
7. Button, submitting the form with text “**Add Offer**”

All fields are **required**. Use a convention for name attributes: **lower\_snake\_case**.

# Structure

You are required to have index.php in the project, even if it is initially empty. Consider all **object oriented best practices** you have been taught during the course.

# Initial File Structure

1. Create a file named “export.sql” holding CREATE DATABASE and all the CREATE TABLE statements. Put it in the top-level directory;
2. Create a directory named “Config” in the top-level directory. Put file named “db.ini” holding the database connection information under the following keys: dsn, user, pass;
3. Create endpoints (front layer) for all possible pages in the top-level directory: index, register, login, add\_offer, edit\_offer, all\_offers, my\_offers, view\_offer and logout;
4. Extract common operation such as autoloading classes in “common.php”.

# Database Abstraction Layer

Create an **abstraction layer over the database** operations in three stages – **prepare -> execute -> take result**. All objects should be in the \Database namespace.

# Repository Layer

Create repository layer in the namespace \App\Repository

# Service Layer

Create service layer in the namespace \App\Service

# Booking Application

Make **all the layers to work**, according to the structure and the business requirement in **1.** and submit the **zip archive** in judge.

# Bonus points

For bonus points you must to the following:

- **Use lazy loading**

- **Days** **must** be **integer** and **bigger** or **equal** to **1**

- **Price must** be **positive number**

- **Password** must contains **one** or **more** **lower** case, **upper** case, **number** and **special** char **EACH.**

Special chars are: !**@#$%^&\*()-=**

Example**:** qwertY1$

- **Email** must be: **<user>**@**<host>**

**<user>** is a sequence of **letters** and **digits**, where '.', '-' and '\_' can appear between them.

-Examples of valid users: "**stephan**", "**mike03**", "**s.johnson**", "**st\_steward**", "**softuni-bulgaria**", "**12345**".

-Examples of invalid users: ''**--123**", "**.....**", "**nakov\_-**", "**\_steve**", "**.info**".

**<host>** is a sequence of at least two words, separated by dots '**.**'. Each word is sequence of letters and can have hyphens '**-**' between the letters.

Examples of hosts: "**softuni.bg**", "**software-university.com**", "**intoprogramming.info**", "**mail.softuni.org**".

Examples of invalid hosts: "**helloworld**", "**.unknown.soft.**", "**invalid-host-**", "**invalid-**".

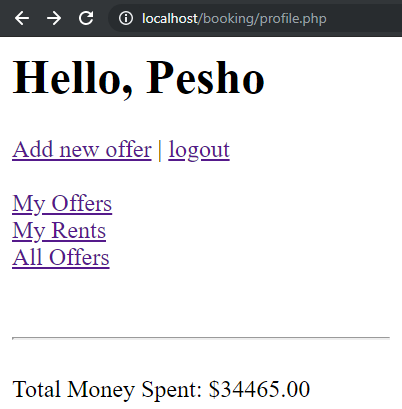
- **Phone** must be like these:

089 955 5592, +359 899 555 592, 0899555592

**6** or **more** digits, **can** start with **+** (but can’t have + anywhere else), **can** have spaces.

**Not** valid phones: 089955559+2, @213-123-1231, 089 955 5592a

* **\*\*Show all rents by user** - **users\_rents.php**



Print information about all rented rooms separated with <hr />

