# SoftUni Forum Workshop - Part 3

This application is created as a workshop for the [Angular Corse](https://softuni.bg/trainings/3047/angular-november-2020) @ [Software University](https://softuni.bg)

# Task Requirements

At this point the application has **forms**. Now it's the time to edit them so the given information is handled properly as expected. Implement the following **validation**, **notification** and **styling** **rules** for each form.

By default:

* The input fields are **neutral** (neither valid nor invalid). If some of the fields are **touched**, you should determinate the status of validation from the given data inside it
* Each input field is **required**

**Notifications** should be shown as paragraph ("**<p>**") elements **below** each input field or button.

The notification **below each button** should display the **status** of the **form validation**:

* If the status is **untouched** or **valid** the **left border** of the input field must be **green**, else **red**.
* If the status is **invalid**, use - "**You can't continue until you provide a valid data**" as text or the default text provided from the skeleton. **The color** inside that paragraph should be "**red**" and the **button** should be **disabled**.
* If the status is **valid** just **remove** the **notification** and **enable** the **button**.

Check the examples below each form for more clarity.

## Invalid Cases

In case some of the input fields аrе invalid, the following **input field** should be styled with class **input-error**, which has the following style:

* "**border-left-color:** **red;**"

And **notification** should be shown with information about the validation. The text inside that paragraph also should be **red**.

## Valid Cases

When the input field is filled with **valid** information, the class **input-error** should be **disabled / removed** so the left border should be **green** again.

## Reactive Forms

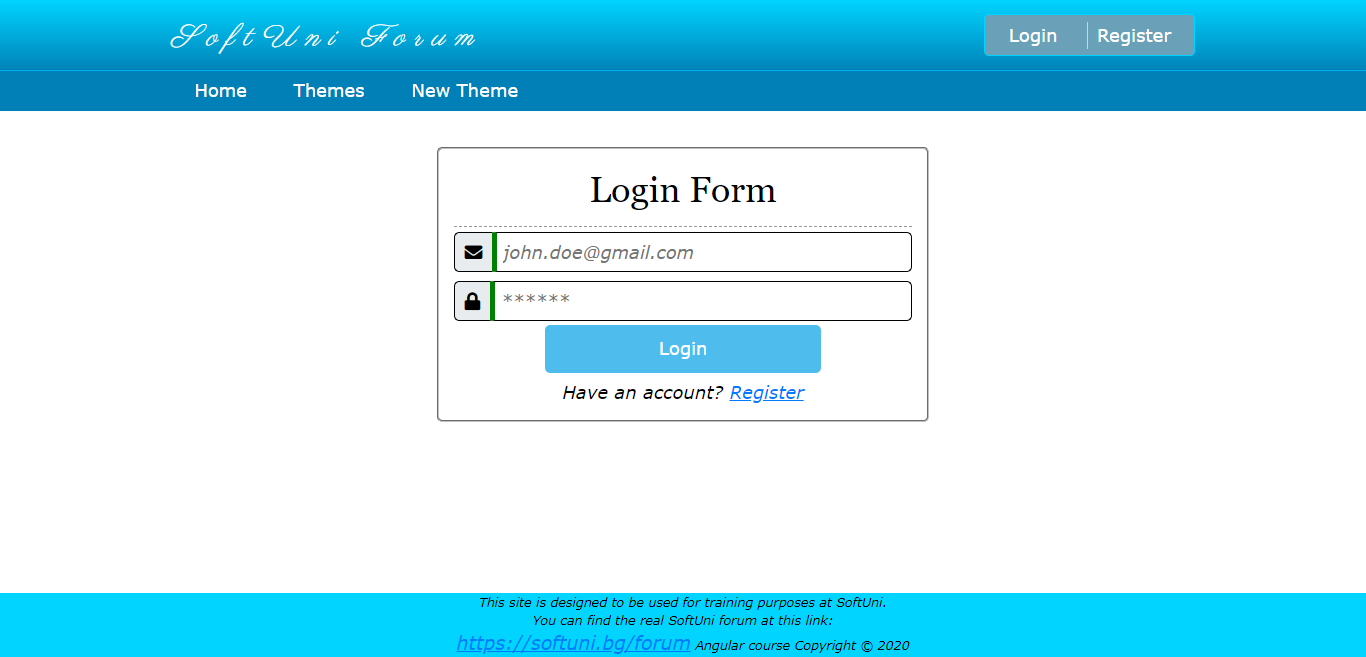
Use the **model driven (reactive)** **approach** to build **login** and the **register** form.

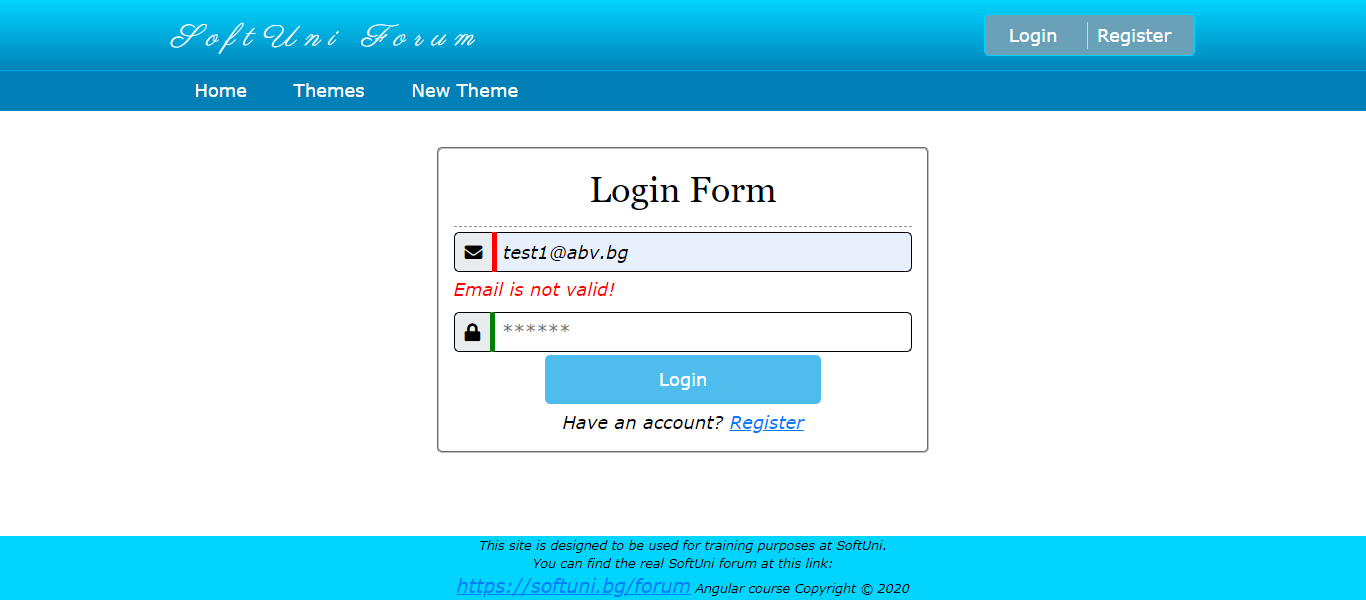
### Login

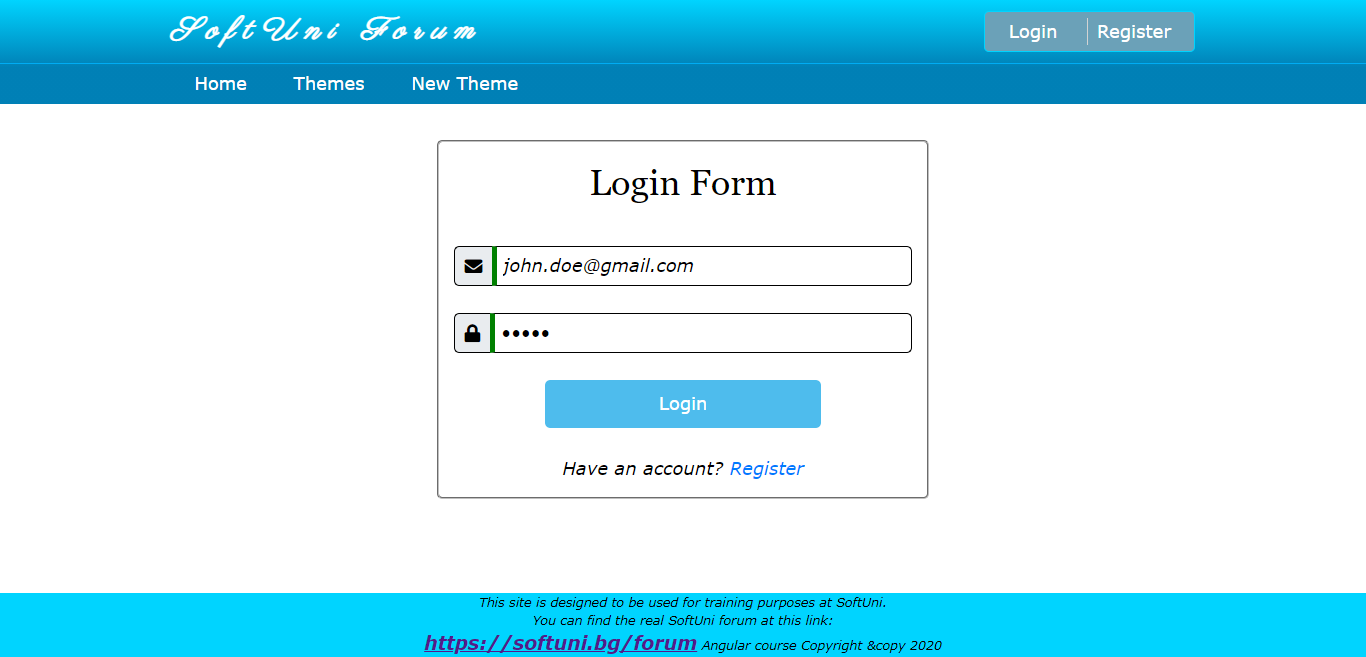
The login form expects as input: **email** and **password**.

Valid credentials in this case are:

* **Email address** must have at least **6 symbols**, after that the symbol "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [something.else@gmail.com](mailto:something.else@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)
* **Password** should be at least **5 symbols** as well







### Register

The register form expects as input: **username**, **email**, **password, repeat password** and **telephone number** (not required).

**Username validation** here is:

* Should have at least **5 symbols** and is **required**.

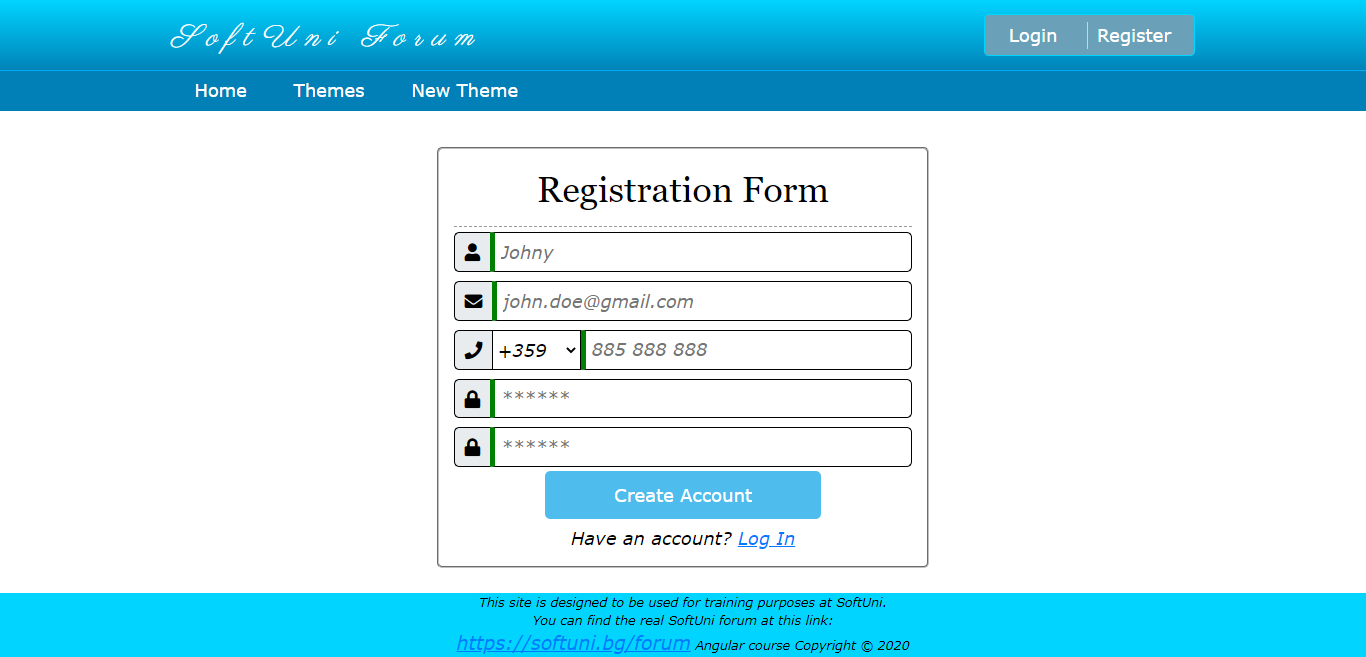
**Email validation** here is the same as the login case:

* Should have at least **6 symbols**, after that the symbol of "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [something.else@gmail.com](mailto:something.else@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)

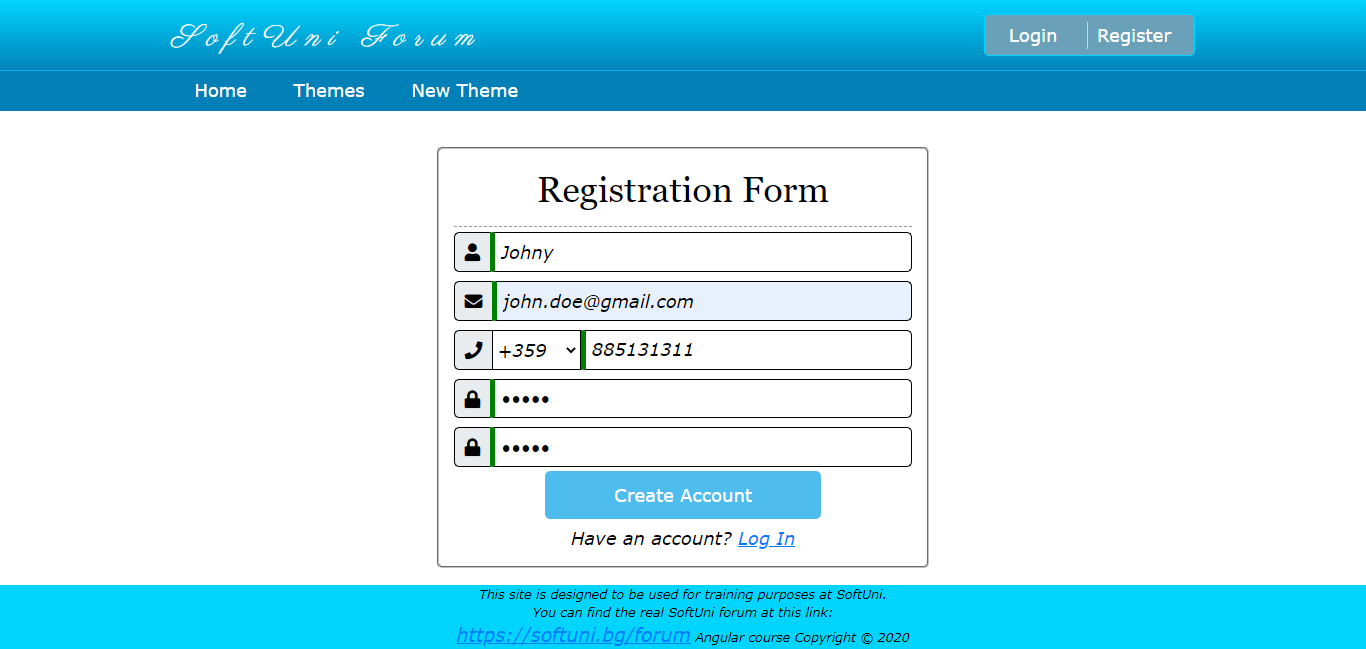
**Both passwords** should be grouped in "**passwords**" group. The validation here is:

* The given **password** should be at **least 5 characters** (any English letter and digits are allowed)
* The given **repeat password should** be the **same** as the given **password**

**Both fields** are considered **invalid** if some of them **do not cover** the described validation. If that happens apply the **invalid** **stylization** for **both**. Otherwise the **valid** **stylization** is applied.







## Template-driven Forms

Use the **template**-**driven** **approach** to build the **create new theme** and the **change user information** form.

### New Theme

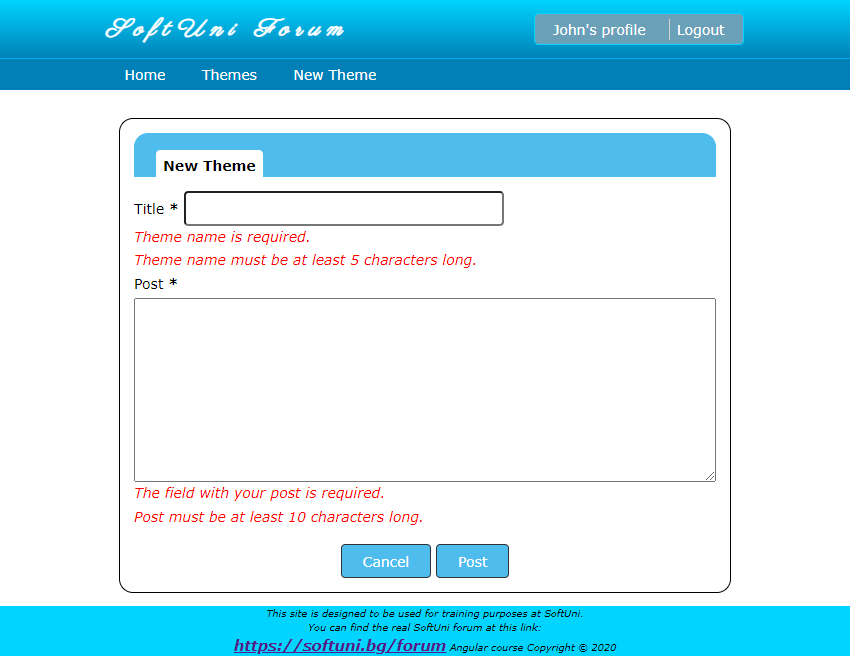
The new theme form expects as input: **theme name**, **post text**.

Valid data in this case is:

* Theme name - should be at **least 5 characters**
* Post text - should be at **least 10 characters** long

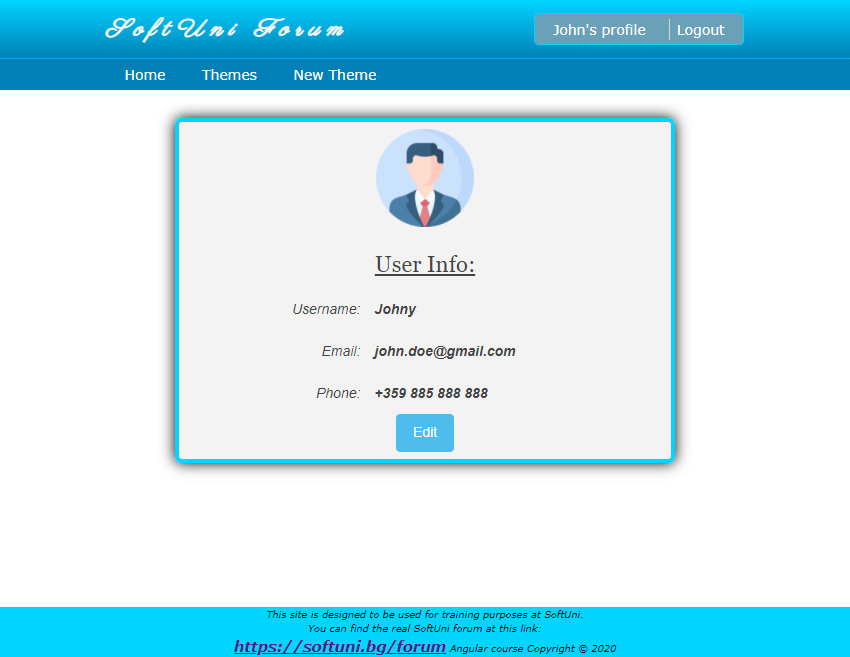
The **"Post"** button should be **disabled** if the inputs are **NOT** valid.



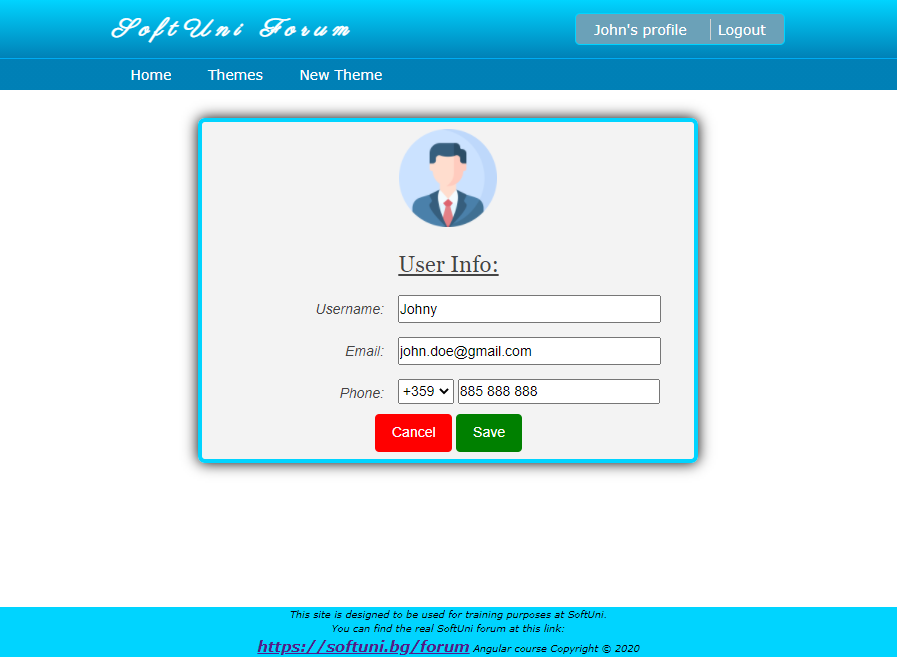


### Change Profile Information

When you are logged in you can see your **profile information**.



By clicking the "**Edit**" button, the profile information should be turned into **form with editable fields**. The input fields should have the current information of the user.



**Username validation** here is the same as the register case:

* Should have at least **5 symbols** and is **required**.

**Email** validation is the same as the login and register cases:

* Should have at least **6 symbols**, after that the symbol of "**@**" is followed. The valid **domain** **name** is only "**gmail**" again followed by "**.**". **Top level** domain can be "**bg**" or "**com**".
  + **Valid** email addresses: [testing@gmail.com](mailto:testing@gmail.com), [testing@gmail.bg](mailto:testing@gmail.bg), [somethingelse@gmail.com](mailto:somethingelse@gmail.com)
  + **Invalid** email addresses: [qwe@gmail.com](mailto:qwe@gmail.com), [123@gmail.bg](mailto:123@gmail.bg), [somethingelse@abv.bg](mailto:somethingelse@abv.bg), [another@abv.com](mailto:another@abv.com)

If the validation passes, the "**Save**" button should be enabled, and the user can make a **post** request to change the information.

If the "**Cancel**" button is clicked all changes must be **reverted**, and the user information should be displayed **unchanged**.

## Custom Directives

Think about for a second which logic can be isolated in custom directive so you avoid repeating part of the code.

**Hint: Stylization / Validation**

***GOOD LUCK*!**