# Exercises: HTTP Protocol

Problems for exercises and homework for the [“Java Web Development Basics” course @ SoftUni](https://softuni.bg/trainings/1284/java-web-development-january-2016).

In this homework, you are supposed to **write java programs** that are run by a web server. You will learn how to create **HTML** content and how to use basic **HTTP** **GET** and **POST** methods. Moreover, you will have to store data in different data structures and write simple algorithms.

**Before you begin make sure you are have followed the tutorial “” on how to configure Apache Web Server.**

## By the Cake: add\_cake.cgi

Extend your web site. Create a CGI script called add\_cake.cgi. Write another program which will be responsible for adding cakes functionality. It should have a simple form to add cakes. **Create a Cake class**. Each cake should have name and price. When you click the submit button a new cake should be created and saved in a list. The newly created cake should be printed below the form. **“name”** and **“price”** are the parameters of the request.

**In order to work create a functionality which processes POST requests.**

### Examples



### Hint

Don’t forget to add CGI parameters during invocation of your java program.

-Dcgi.query\_string=$QUERY\_STRING

-Dcgi.request\_method=$REQUEST\_METHOD

## By the Cake: Write data

Extend your web site. The submitted data should be appended on a new row split by comma in a file called **database.csv**. Add a link to go back to your main page.



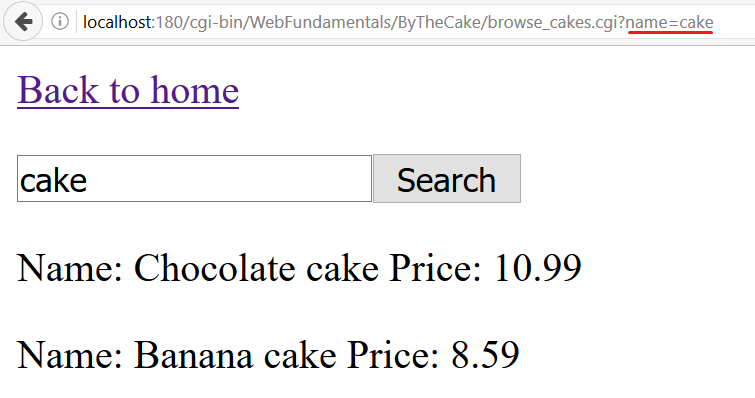
## By the Cake: browse\_cakes.cgi

Extend your web site. Crate a new program that should print a single form **with GET method**. See the example below. It will be used to search cakes by name. Cakes should be searched from **database.csv** and printed below the form.

Add a functionality to go back to your main page.

You will need a third CGI script called browse\_cakes.cgi to invoke your program.

### Examples

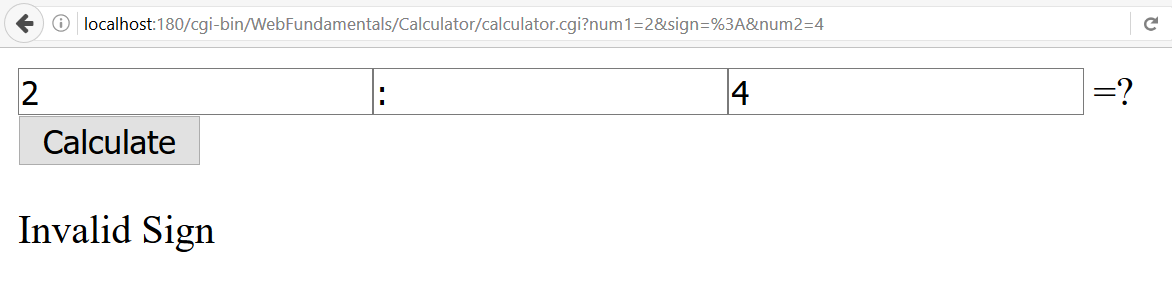
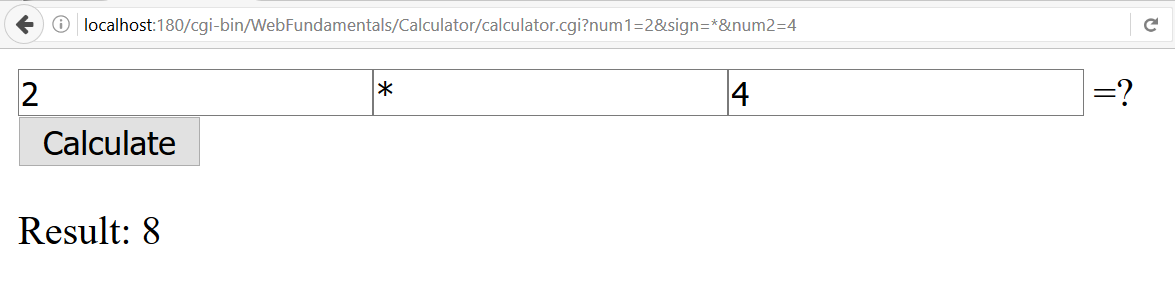
**database.csv***Chocolate cake,10.99  
Banana cake,8.59* ****

## Calculator

Write a java program that prints a form with three inputs on the browser followed by single button. Input 1 and input 3 should receive numbers. Input 2 should receive a mathematical sign. When you press the button the result of the equation should appear. See the example below. There are four possible operations: add, subtract, multiply and divide (+, -, \*, /). If none of the above is entered print an error “Invalid Sign”.

**Use GET method.**

### Examples



## Log in form

Write a java program that prints a form with two inputs on the browser followed by а single button. Inputs should receive username and password. When you click the button a message saying “Hi {username}, your password is {password}” should appear.

**Use POST method.**

### Examples



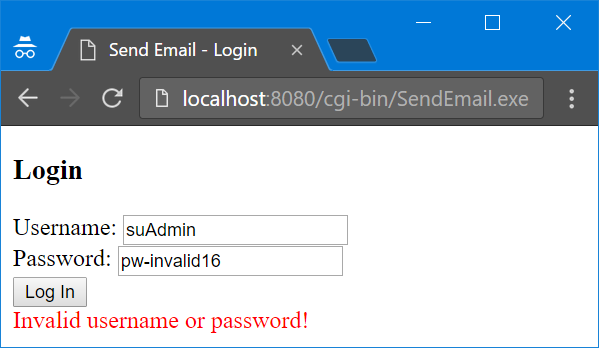
## Send Email

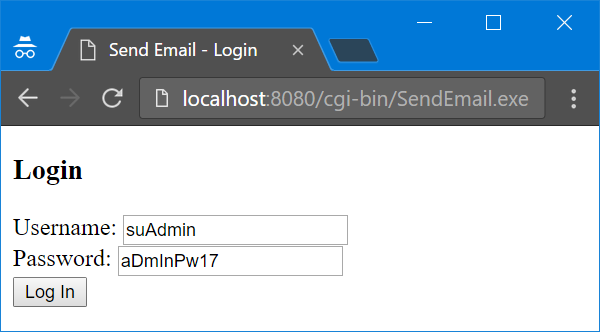
Write a Java program that prints a form with 2 text fields for **username** and **password**. A user must be able to log in only if the **provided username** is suAdmin and the **password** is aDmInPw17. When is logged in he/she should be able to **send email via provided form** where recipient email, subject and body of the email must be provided. Before sending an email validate the provided data:

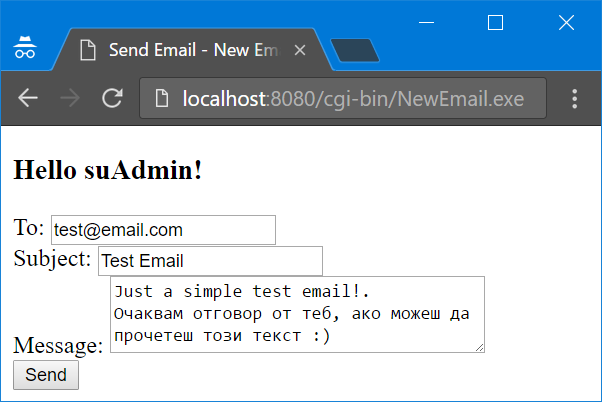
* **To** - must be a **valid email** (search in the internet for the requirements of an email address)
* **Subject** - must be no longer than **100 symbols**
* **Message** - must be able to send **all Unicode characters**

The email messages must be sent from **your own email address**.

### Example



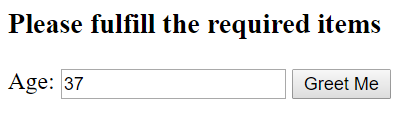
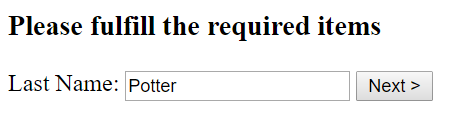




## Greetings

Write a program that collects some data from the user and finally prints a greeting message. Check the examples below.

### Example



## Survey

Write a Java program that creates a **survey for receiving feedback from users for a given product**. Look at the example to check what fields the feedback form should have and how they should be styled. In the **Status** field the possible options are:

* Student
* Part-time Employee
* Full-time Employee
* Unemployed
* Do not want to answer

Make **appropriate validations** where it is necessary.

When the form is submitted **store each submission** in a file called **survey-results.csv**

### Example

