CSC 405
Computer Security

Web Security

Alexandros Kapravelos
akaprav@ncsu.edu

(Derived from slides by Giovanni Vigna and Adam Doupe)

HTML

- Original HTML had
  - images
  - tables
  - font sizes
  - …
- Content was static

Yahoo! Deutschland  Weekly Picks

Search Options

Yellow Pages - People Search - City Maps -- News Headlines - Stock Quotes - Sports Scores

- **Arts** - - *Humanities*, *Photography*, *Architecture*, *...*
- **Business and Economy [Xtra!]** - - *Directory*, *Investments*, *Classifieds*, *...*
- **Computers and Internet [Xtra!]** - - *Internet*, *WWW*, *Software*, *Multimedia*, *...*
- **Education** - - *Universities*, *K-12*, *Courses*, *...*
- **Entertainment [Xtra!]** - - *TV*, *Movies*, *Music*, *Magazines*, *...*
- **Government** - - *Politics [Xtra!]*, *Agencies*, *Law*, *Military*, *...*
- **Health [Xtra!]** - - *Medicine*, *Drugs*, *Diseases*, *Fitness*, *...*
- **News [Xtra!]** - - *World [Xtra!]*, *Daily*, *Current Events*, *...*
- **Recreation and Sports [Xtra!]** - - *Sports*, *Games*, *Travel*, *Autos*, *Outdoors*, *...*
- **Reference** - - *Libraries*, *Dictionaries*, *Phone Numbers*, *...*
- **Regional** - - *Countries*, *Regions*, *U.S. States*, *...*
- **Science** - - *CS*, *Biology*, *Astronomy*, *Engineering*, *...*
- **Social Science** - - *Anthropology*, *Sociology*, *Economics*, *...*
- **Society and Culture** - - *People*, *Environment*, *Religion*, *...*

Yahoo! New York - Yahoo! Shop - Yahooligans!

Yahoo! Japan - Yahoo! Internet Life - Yahoo! San Francisco

source: https://web.archive.org/web/19961017235908/http://www2.yahoo.com/

Welcome to Amazon.com Books!

One million titles, consistently low prices.

(If you explore just one thing, make it our personal notification service. We think it's very cool!)

Spotlight! -- August 16th

These are the books we love, offered at Amazon.com low prices. The spotlight moves **EVERY** day so please come often.

One Million Titles

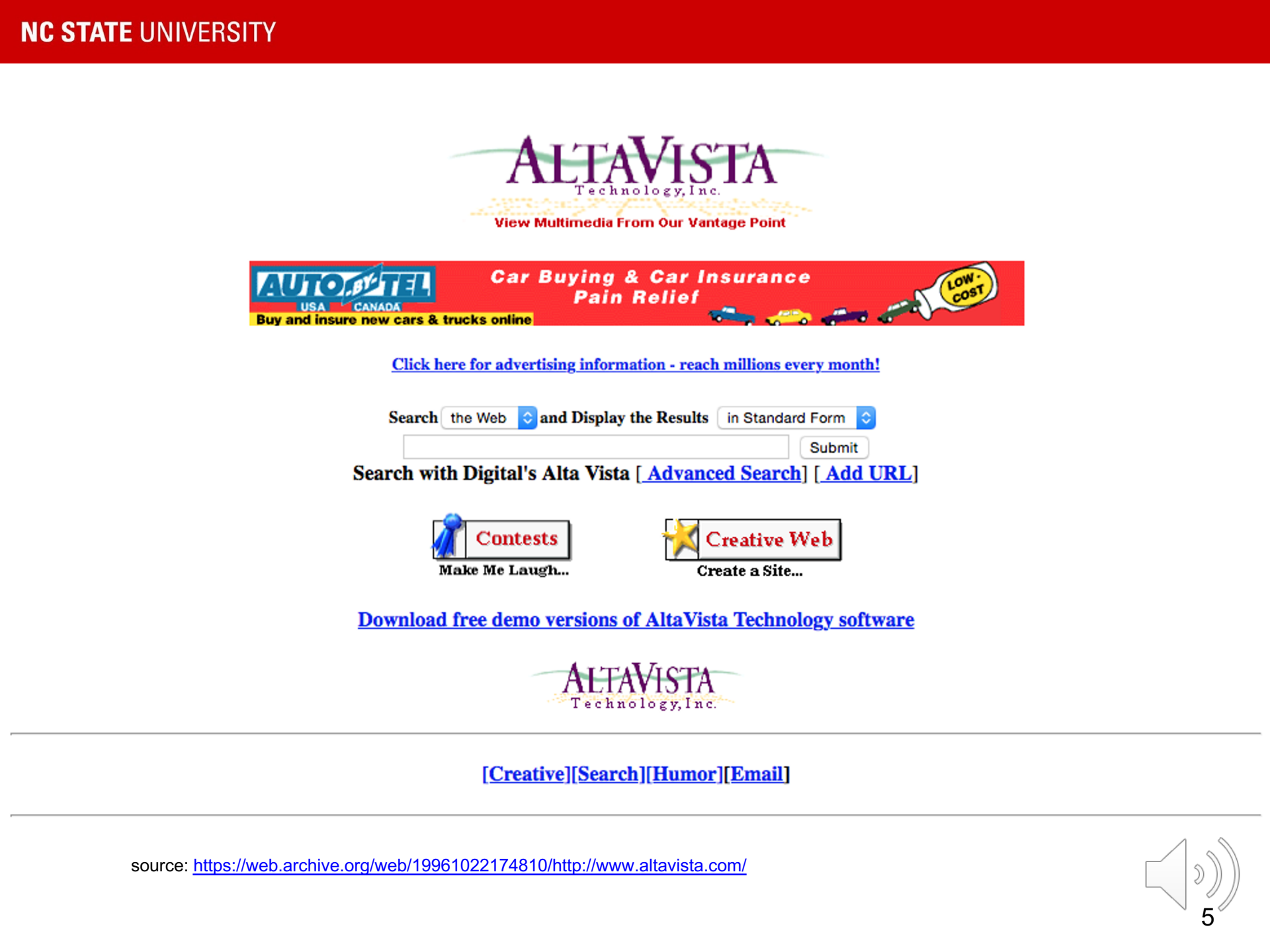
Search Amazon.com's million title catalog by author, subject, title, keyword, and more... Or take a look at the books we recommend in over 20 categories... Check out our customer reviews and the award winners from the Hugo and Nebula to the Pulitzer and Nobel... and bestsellers are 30% off the publishers list...

Eyes & Editors, a personal notification service

Like to know when that book you want comes out in paperback or when your favorite author releases a new title? Eyes, our tireless, automated search agent, will send you mail. Meanwhile, our human editors are busy previewing galleys and reading advance reviews. They can let you know when especially wonderful works are published in particular genres or subject areas. Come in, meet Eyes, and have it all explained.

Your Account

Check the status of your orders or change the email address and password you have on file with us. Please note that you **do not** need an account to use the store. The first time you place an order, you will be given the opportunity to create an account.



source: https://web.archive.org/web/19961022174810/http://www.altavista.com/

source: https://web.archive.org/web/19981202230410/http://www.google.com/

HTML Design

- HTML designed to describe a text document with hyperlinks to other documents
- How to do fancy animations or pretty web pages?

JavaScript

- Client-Side scripting language for interacting and manipulating HTML
- Created by Brendan Eich at Netscape Navigator 2.0 in September 1995 as "LiveScript"
- Renamed to "JavaScript" in December 1995 and is (from the Netscape Press Release)
  - "announced JavaScript, an open, cross-platform object scripting language for the creation and customization of applications on enterprise networks and the Internet"
- JavaScript is a (from wikipedia) "prototype-based scripting language with dynamic typing and first-class functions"
  - Does this sound like Java?
- Questions over why the name change
  - Marketing ploy to capitalize on the "hot" Java language?
  - Collaboration between Sun and Netscape?
- By August 1996, Microsoft added support for JavaScript to Internet Explorer
  - Microsoft later changed the name to JScript to avoid Sun's Java trademark
- Submitted to Ecma International for standardization on November 1996
- ECMA-262, on June 1997, standardized first version of ECMAScript

JavaScript

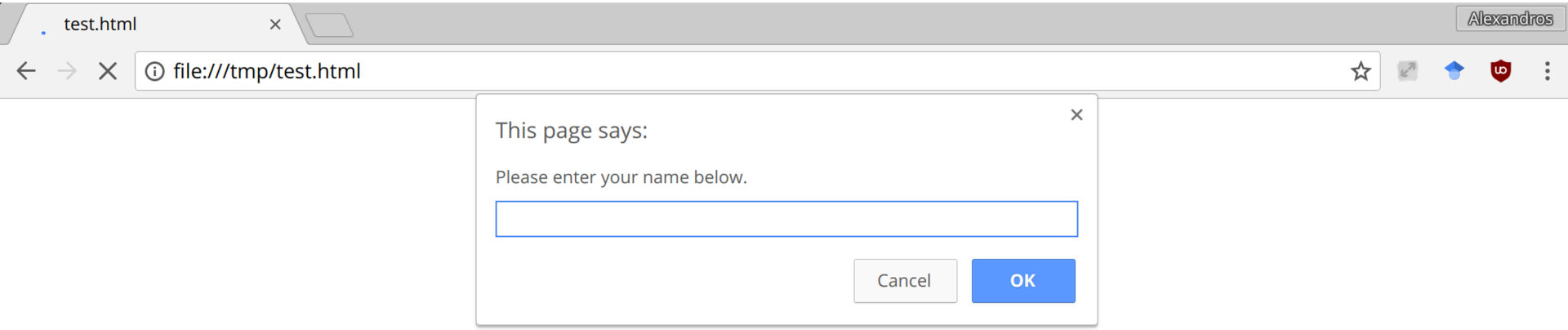
- Lingua franca of the web
- Eventually supported by all browsers
- Language organically evolved along the way

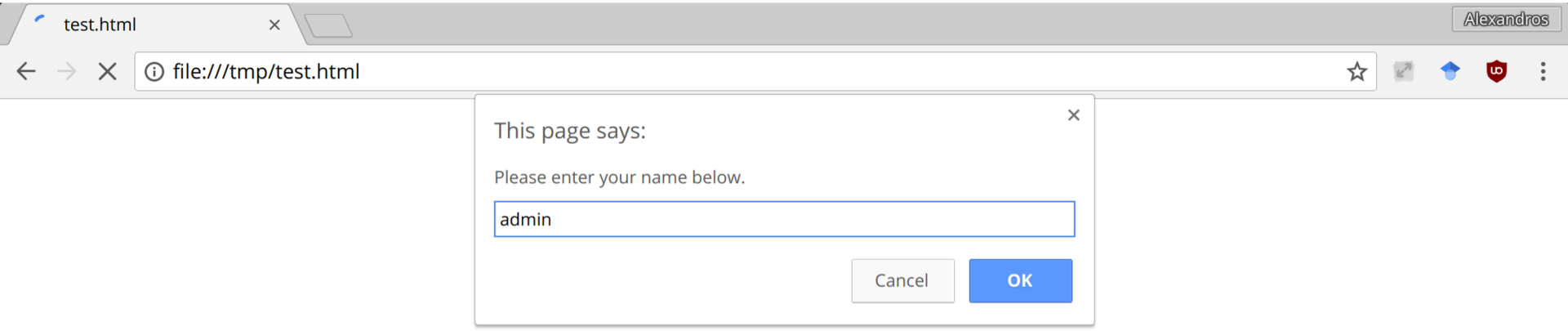
JavaScript

- Code can be embedded into HTML pages using the `script` element and (optionally storing the code in HTML comments)

```
<script>
<!--
var name = prompt('Please enter your name below.', '');
if (name == null) {
  document.write('Welcome to my site!');
}
else {
  document.write('Welcome to my site ' + name + '!');
}
-->
</script>

<script type="text/javascript">
<script language="javascript">
```


test.html
Alexandros
file:///tmp/test.html
This page says:
Please enter your name below.
Cancel
OK


test.html
Alexandros
file:///tmp/test.html
This page says:
Please enter your name below.
admin
Cancel
OK

Welcome to my site admin!

JavaScript

- You can also include external JavaScript files in your HTML
  - As opposed to the inline JavaScript that we saw in the previous example
- `<script src="<absolute or relative URL"></script>`
- When the browser parses this HTML element, it automatically fetches and executes the JavaScript before continuing to parse the rest of the HTML
  - Semantically equivalent as if the JavaScript was directly in the page

Document Object Model (DOM)

- The Document Object Model is a programmatic interface in JavaScript to the manipulation of client-side content
- Created a globally accessible in JavaScript `document` object
  - The document object is used to traverse, query, and manipulate the browser's representation of the HTML page as well as handle events
- DOM 0, released in 1995 with original JavaScript
  - Very basic
- Intermediate DOM began in 1997 with Microsoft and Netscape releasing incompatible improvements to DOM
- W3C stepped in and started to define standards
  - DOM 1, October 1998
  - DOM 2, November 2000
  - DOM 3, April 2004
  - DOM is now a W3C Living Standard, and various snapshots of the standard will turn into DOM4

DOM Example

```
<!DOCTYPE html>
<html>
 <head>
   <meta charset="UTF-8">
   <title>DOM Example</title>
 </head>
 <body>
  <h1>DOM Example</h1>
   <div id='insert_here'>
   </div>
 </body>
 <script>
   var hr = document.createElement('HR');
   document.getElementById('insert_here').appendChild(hr);
 </script>
</html>
```

DOM Example

Using the DOM

- Coding proper DOM access in a cross-browser approach is a nightmare
  - Some highlights from http://stackoverflow.com/questions/565641/what-cross-browser-issues-have-you-faced
    - "Internet Explorer does not replace   or HTML char code 160, you need to replace its Unicode equivalent \u00a0"
    - "In Firefox, a dynamically created input field inside a form (created using document.createElement) does not pass its value on form submit."
    - "document.getElementById in Internet Explorer will return an element even if the element name matches. Mozilla only returns element if id matches."
- jQuery is an amazing library that provides a uniform interface and handles all the DOM cross-browser compatibilities
- Today the situation is match better and compatibility has greatly improved

Browser Object Model (BOM)

- Programmatic interface to everything outside the document (aka the browser)
- No complete standard (the term BOM is colloquial)
- Examples
  - window.name = "New name"
  - window.close()
  - window.location = "http://example.com"

JavaScript vs. DOM and BOM

- JavaScript the language is defined separate from the DOM and BOM
  - DOM has its own specification, and much of the BOM is specified in HTML5 spec
- In the web context, these are often confused, because they are used together so often
- However, now with JavaScript popping up all over the place, it's an important distinction
  - Server-side code using Node.js
  - Database queries (MongoDB)
  - Flash (ActionScript, which has its own DOM-like capabilities)
  - Java applications (javax.script)
  - Windows applications (WinRT)

JavaScript – Object-based

- Almost everything in JavaScript is an object
  - Objects are associative arrays (hash tables), and the properties and values can be added and deleted at run-time

```
var object = {test: "foo", num: 50};
object['foo'] = object;
console.log(object[object['test']]);
object.num = 1000;
console.log(object['num']);
```

```
> var object = {test: "foo", num: 50};
< undefined
> object['foo'] = object;
< ▼ Object {test: "foo", num: 50, foo: Object} i
    ▶ foo: Object
      num: 1000
      test: "foo"
    ▶ __proto__: Object
> console.log(object[object['test']]);
   ▶ Object {test: "foo", num: 50, foo: Object}
< undefined
> object.num = 1000;
< 1000
> console.log(object['num']);
   1000
< undefined
>
```

JavaScript – Recursion

```
function factorial(n) {
    if (n === 0) {
        return 1;
    }
    return n * factorial(n - 1);
}
console.log(factorial(5));
120
```

http://en.wikipedia.org/wiki/JavaScript

JavaScript – Anonymous Functions and Closures

```
var createFunction = function() {
    var count = 0;
    return function () {
        return ++count;
    };
};
var inc = createFunction();
inc();
inc();
inc();
var inc2 = createFunction();
inc2();
```

```
> var createFunction = function() {
      var count = 0;
      return function () {
          return ++count;
      };
  };
< undefined
> var inc = createFunction();
< undefined
> inc();
< 1
> inc();
< 2
> inc();
< 3
> var inc2 = createFunction();
< undefined
> inc2();
< 1
>
```

JavaScript – Runtime Evaluation

- JavaScript contains features to interpret a string as code and execute it
  - eval
  - Function
  - setTimeout
  - setInterval
  - execScript (deprecated since IE11)

```
var foo = "bar";
eval("foo = 'admin';");
console.log(foo);
var x = "console.log('hello');";
var test = new Function(x);
test();
```

```
> var foo = "bar";
< undefined
> eval("foo = 'admin';");
< "admin"
> console.log(foo);
  admin                                          VM49:1
< undefined
> var x = "console.log('hello');";
< undefined
> var test = new Function(x);
< undefined
> test()
  hello                                          VM54:2
< undefined
>
```

JavaScript Uses – Form Validation

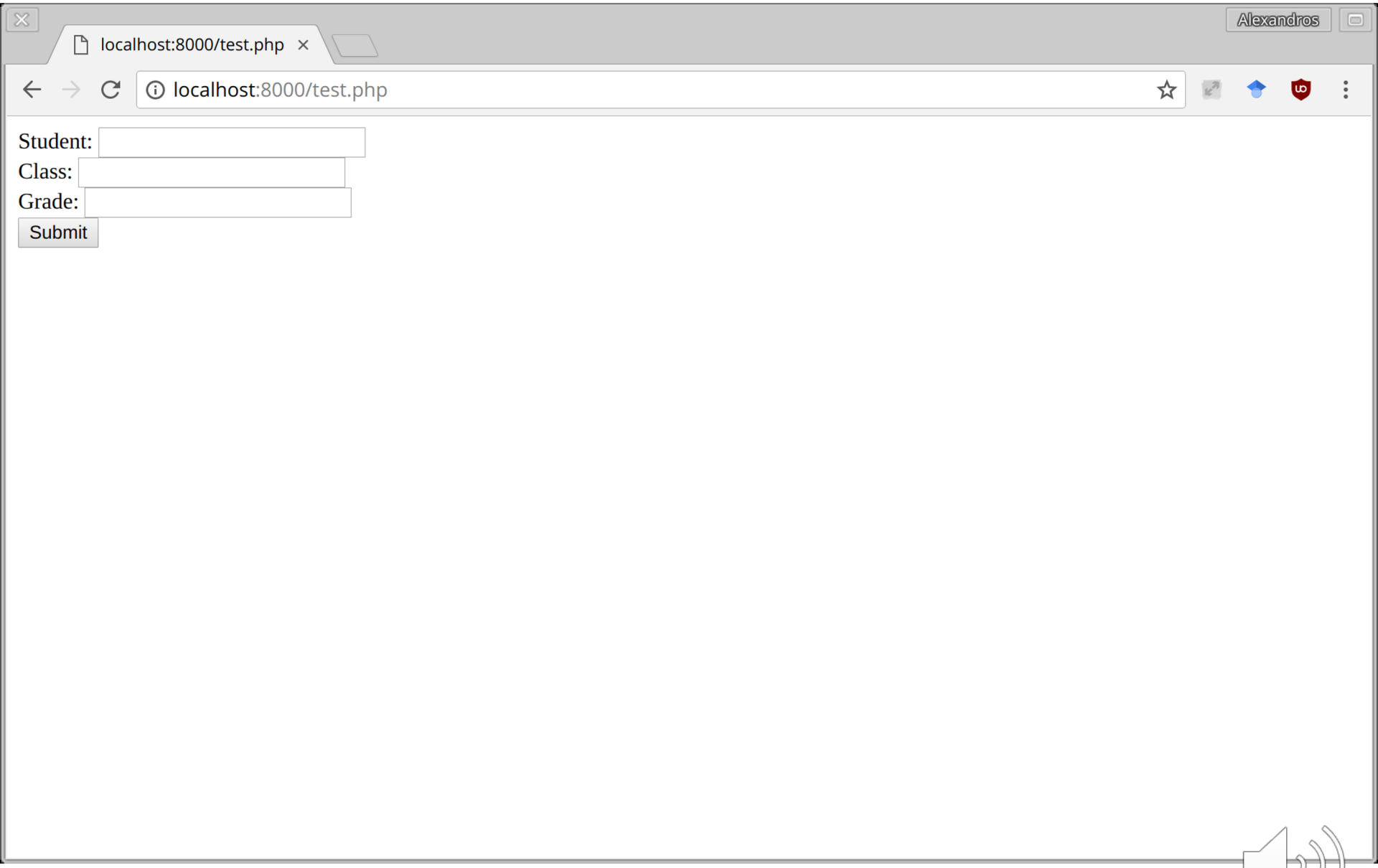
- How to validate user input on HTML forms?
- Traditionally requires a round-trip to the server, where the server can check the input to make sure that it is valid

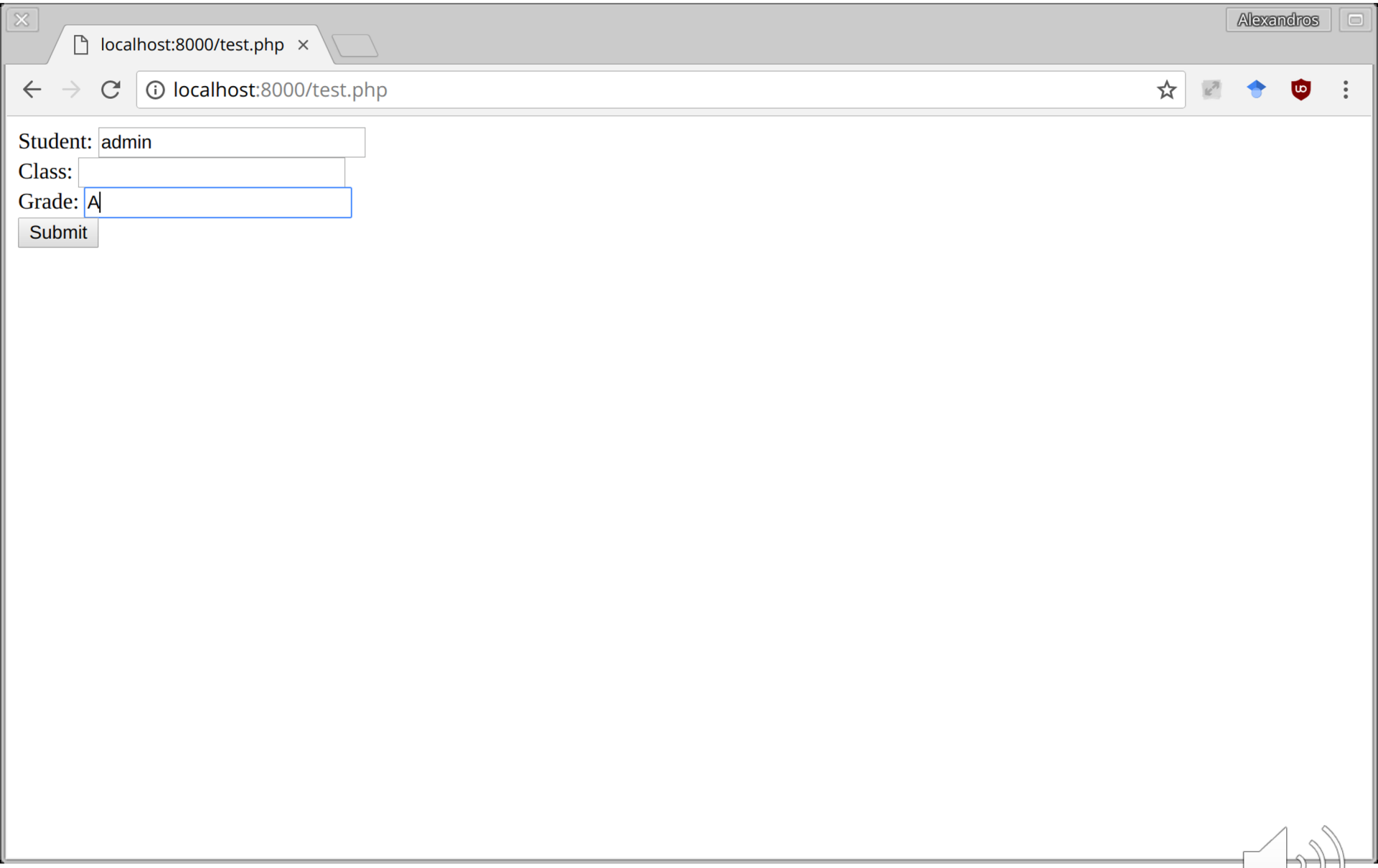
JavaScript Uses – Form Validation

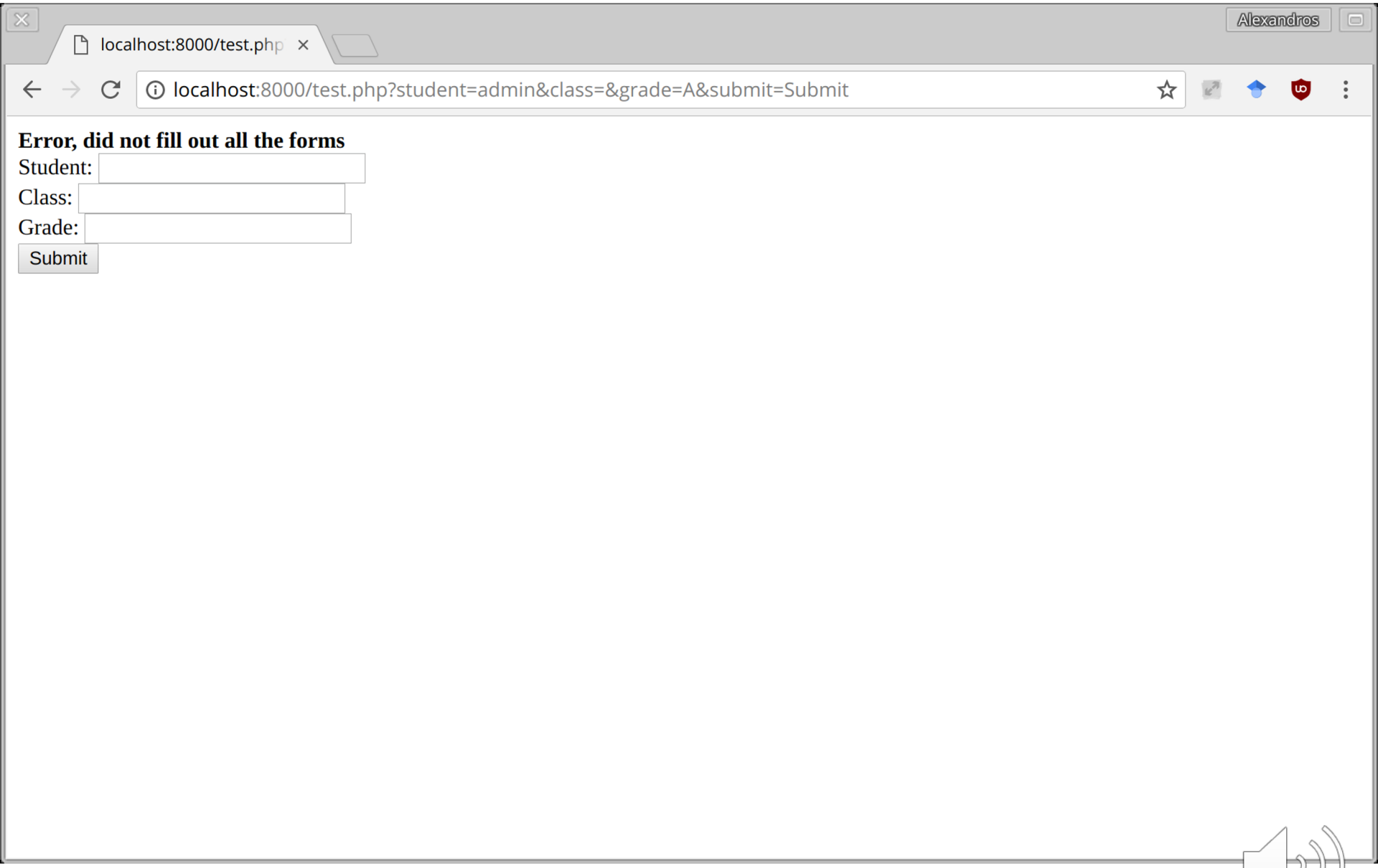
```
<?php
if ($_GET['submit']) {
  $student = $_GET['student'];
  $class = $_GET['class'];
  $grade = $_GET['grade'];
  if (empty($student) || empty($class) || empty($grade)) {
            echo "<b>Error, did not fill out all the forms</b>";
  }
  else if (!($grade == 'A' || $grade == 'B' || $grade == 'C' ||
               $grade == 'D' || $grade == 'F')) {
            echo "<b>Error, grade must be one of A, B, C, D, or F</b>";
  }
  else { echo "<b>Grade successfully submitted!</b>";
  }
} ?>
<form>
 Student: <input type="text" name="student"><br>
 Class:  <input type="text" name="class"><br>
 Grade:  <input type="text" name="grade"><br>
 <input type="submit" name="submit">
</form>
```

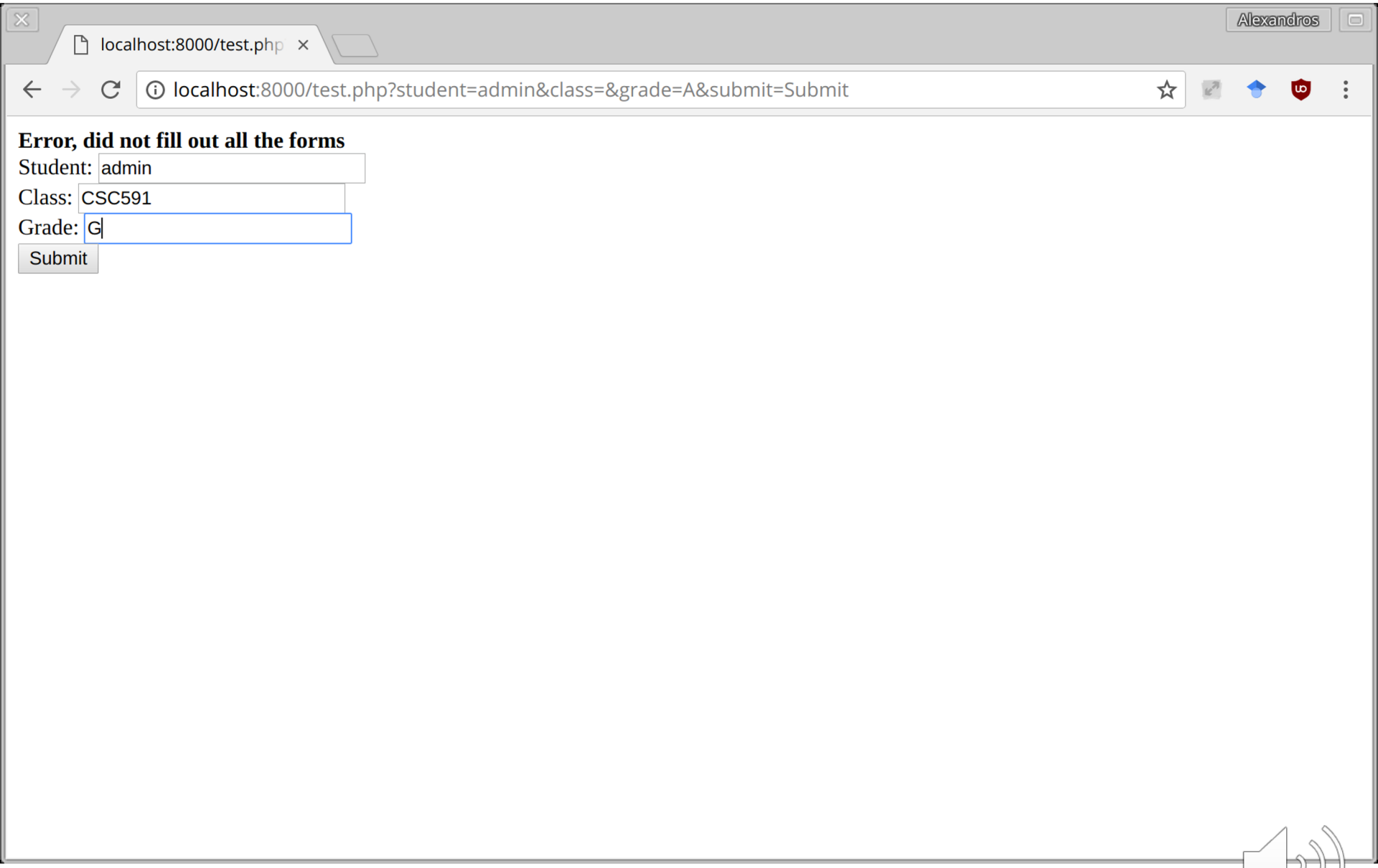
Quick tip:

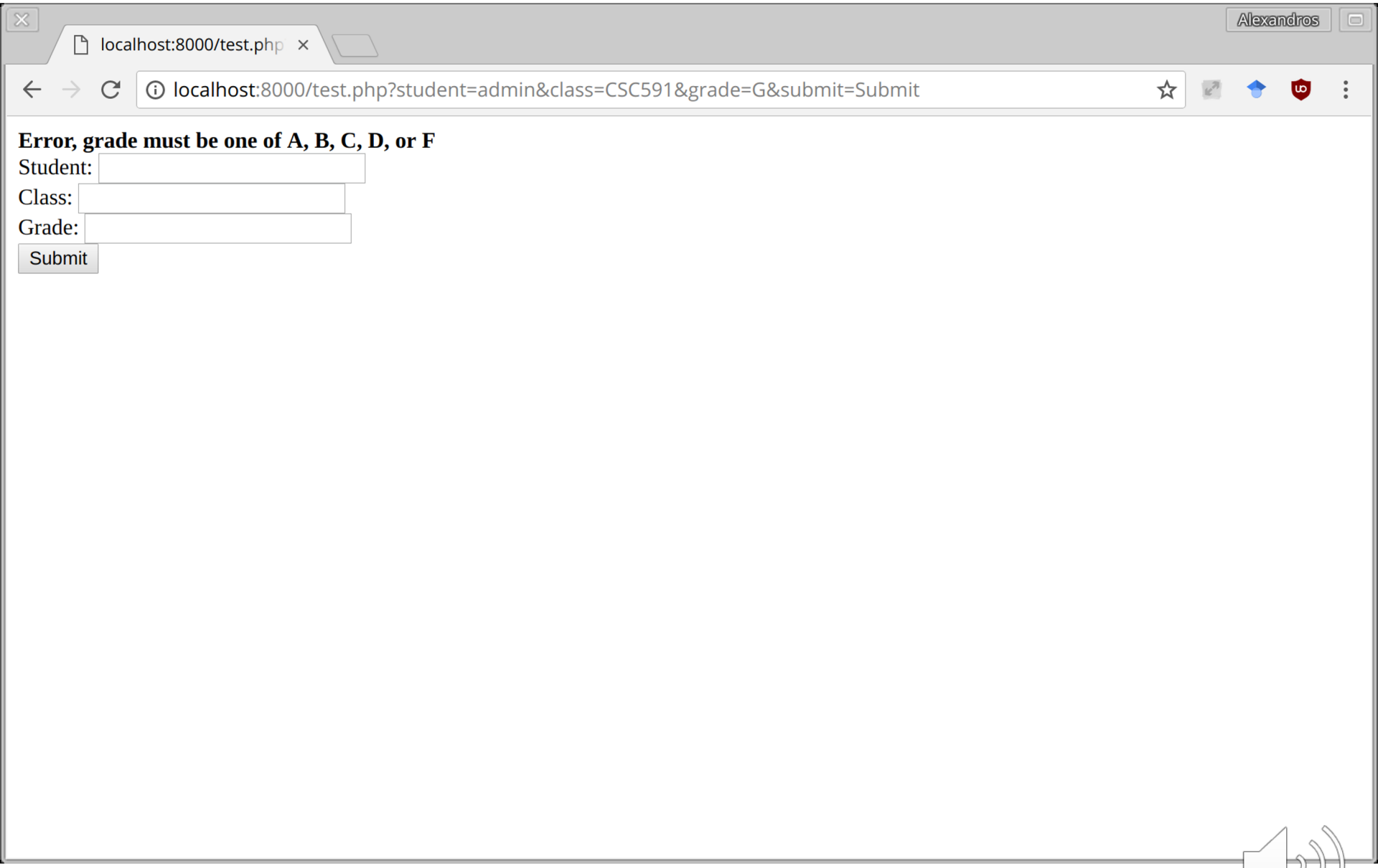
```
$ cd /var/www/public_html
$ php -S localhost:8000
```

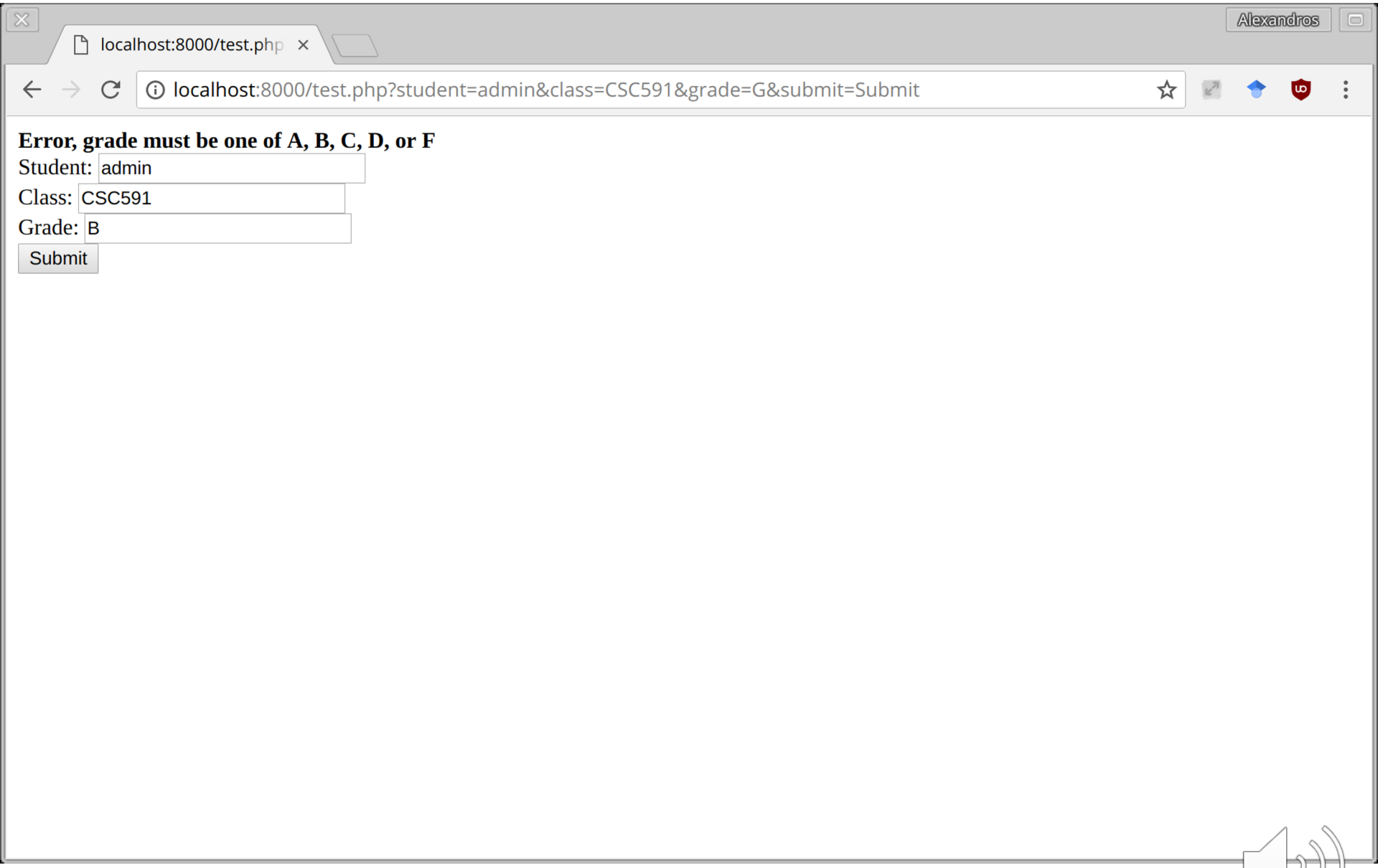

Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student:
Class:
Grade:
Submit

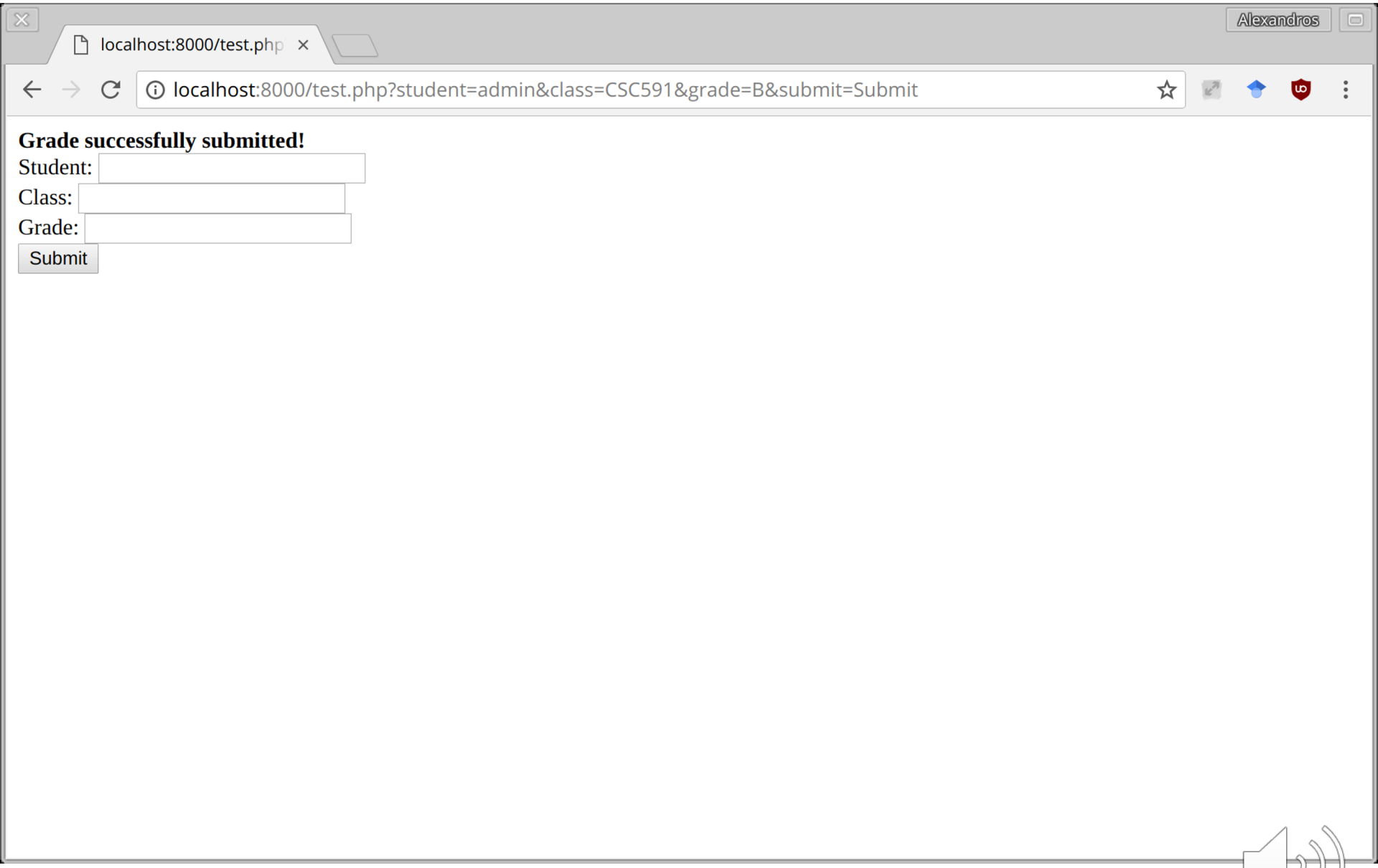

Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class:
Grade: A
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=&grade=A&submit=Submit
Error, did not fill out all the forms
Student:
Class:
Grade:
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=&grade=A&submit=Submit
Error, did not fill out all the forms
Student: admin
Class: CSC591
Grade: G
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=CSC591&grade=G&submit=Submit
Error, grade must be one of A, B, C, D, or F
Student:
Class:
Grade:
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=CSC591&grade=G&submit=Submit
Error, grade must be one of A, B, C, D, or F
Student: admin
Class: CSC591
Grade: B
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=CSC591&grade=B&submit=Submit
Grade successfully submitted!
Student:
Class:
Grade:
Submit

form_validation_regular.php

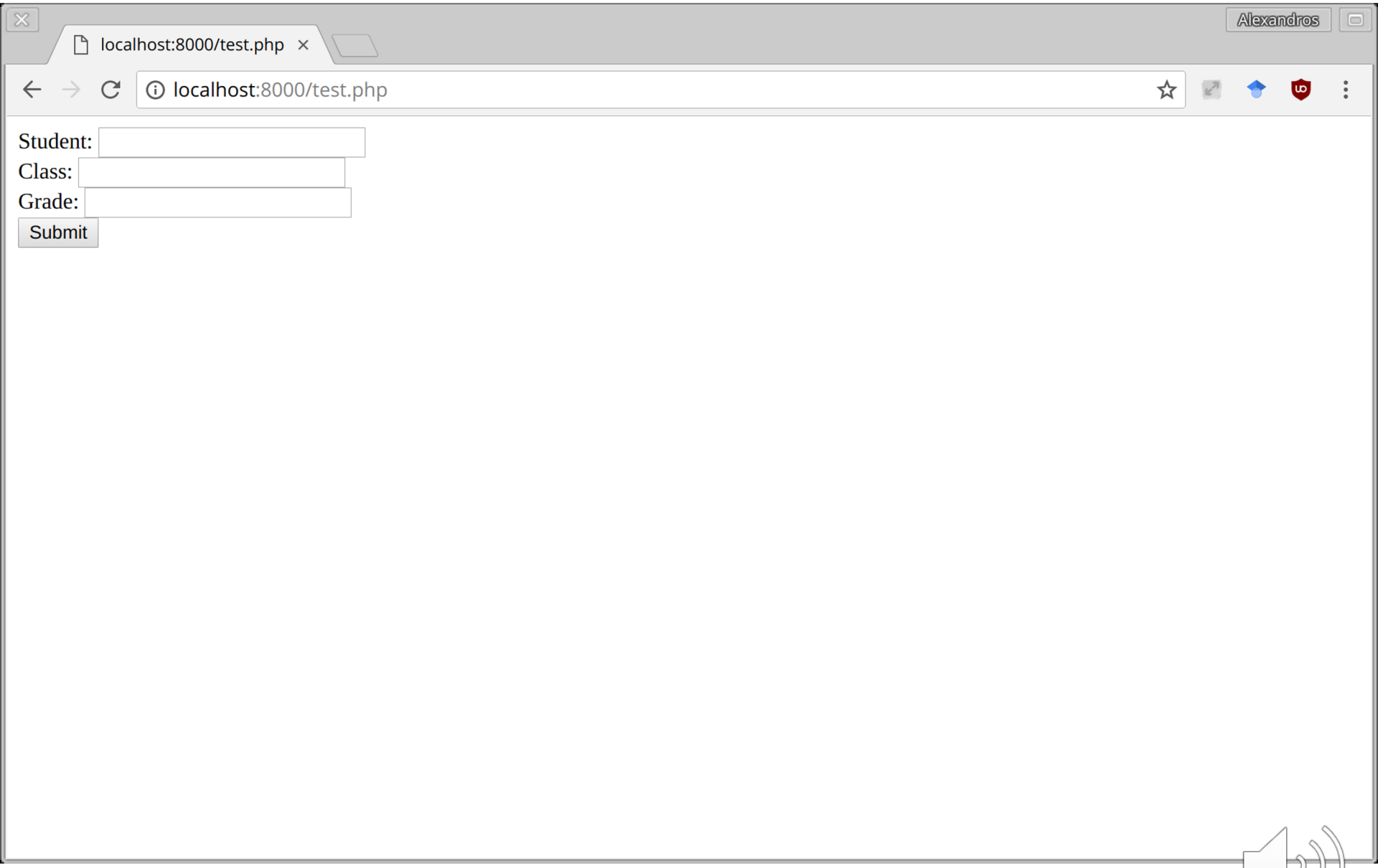
empty class field

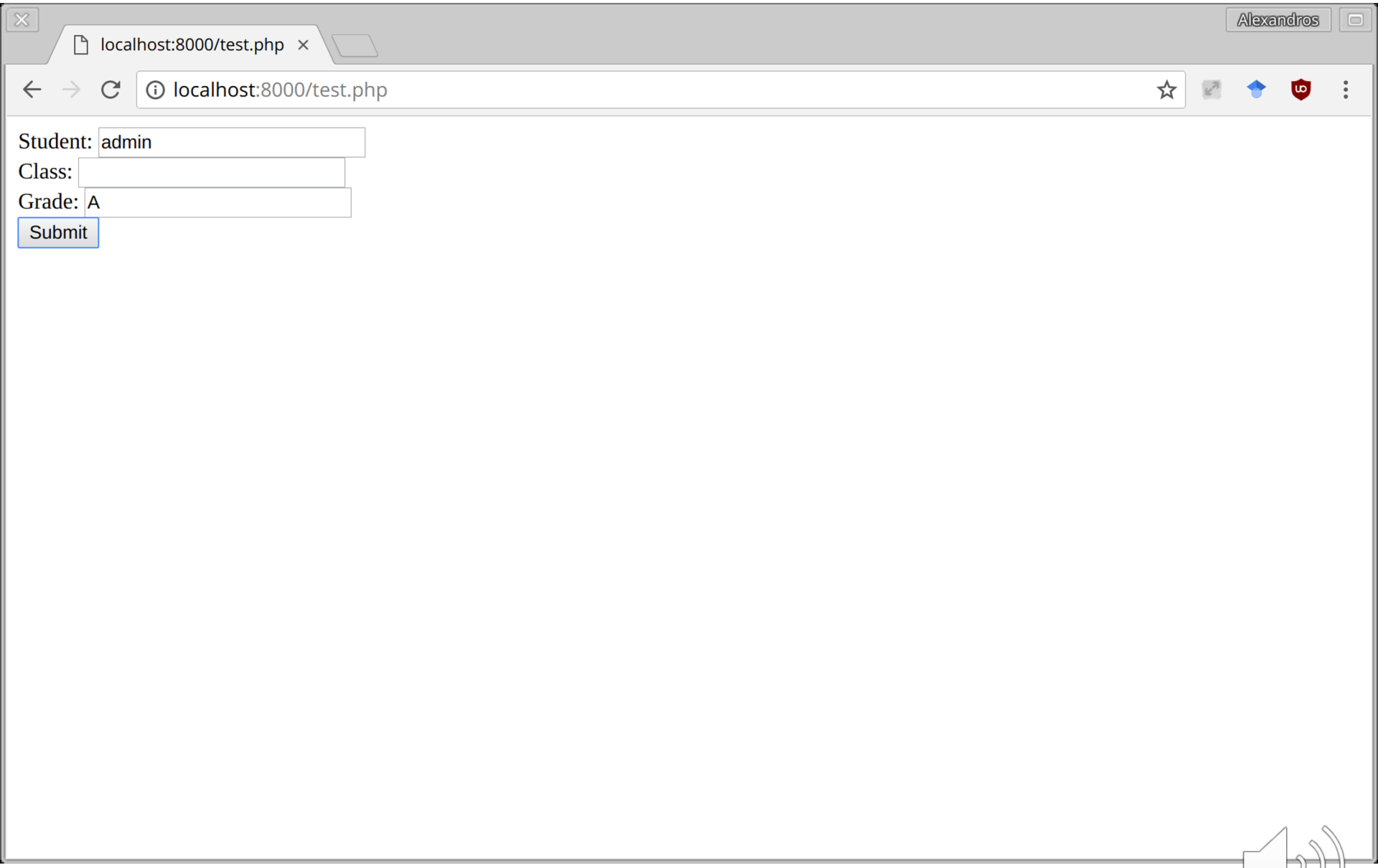
wrong grade format

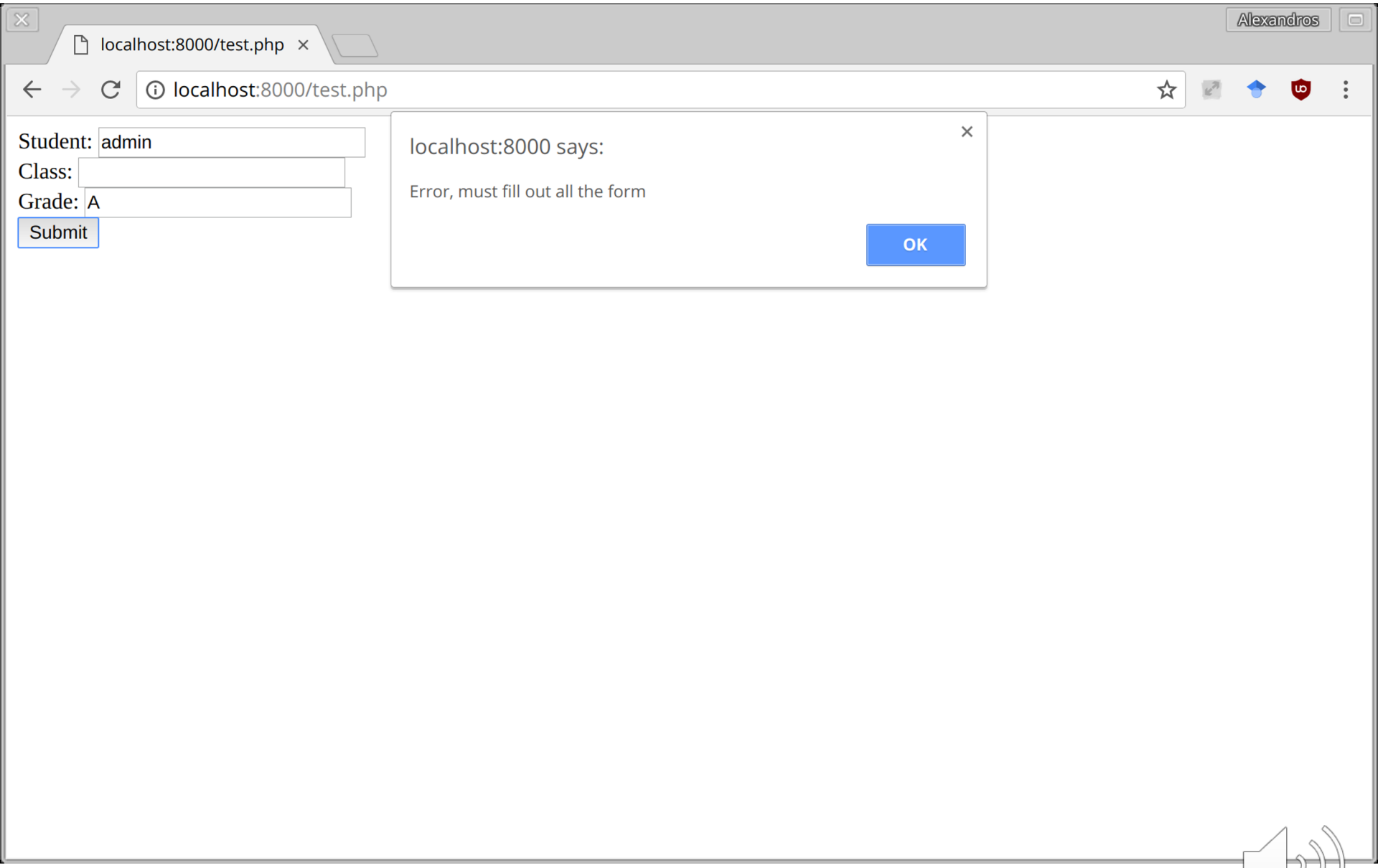
correct submission

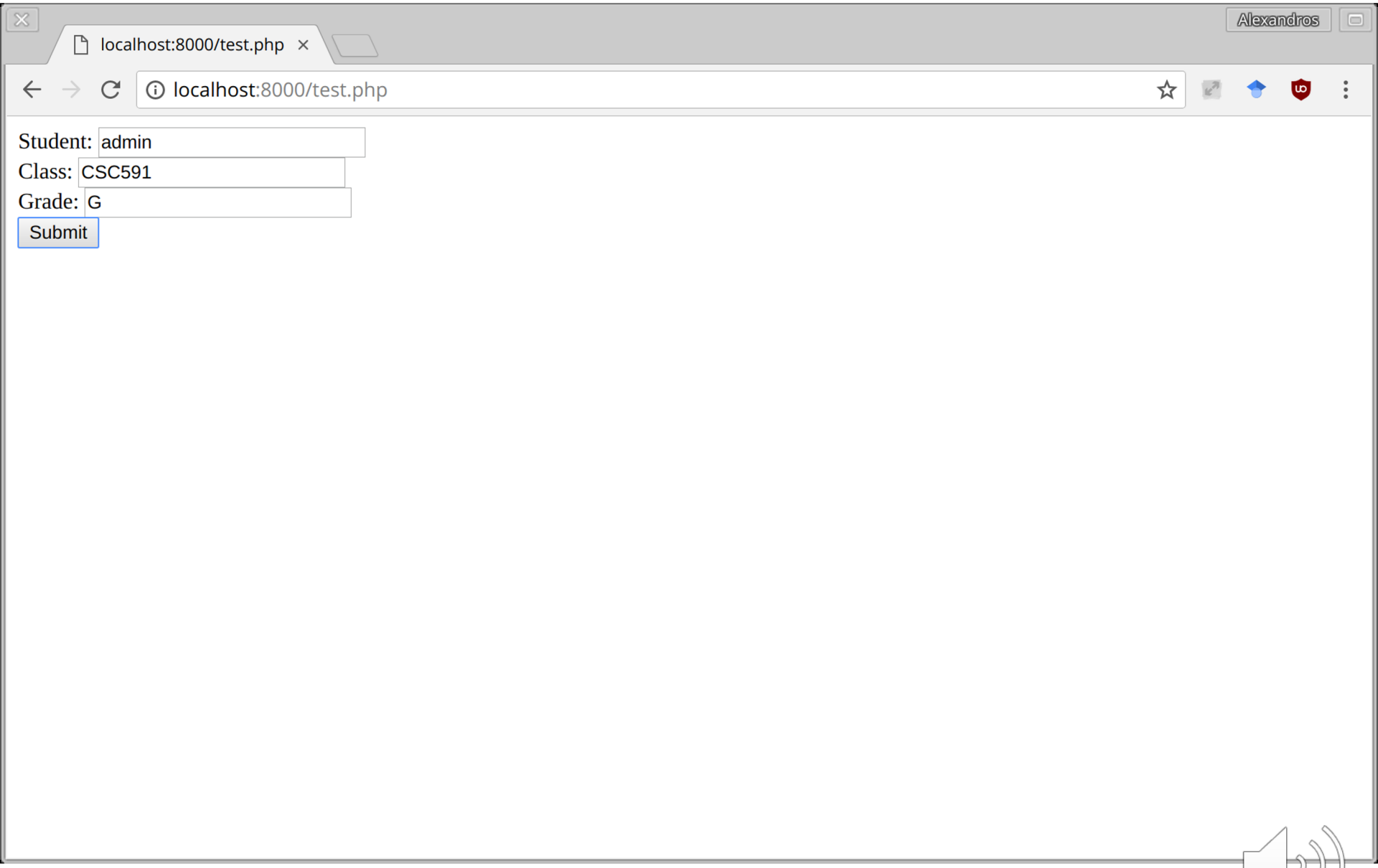
JavaScript Uses – Form Validation

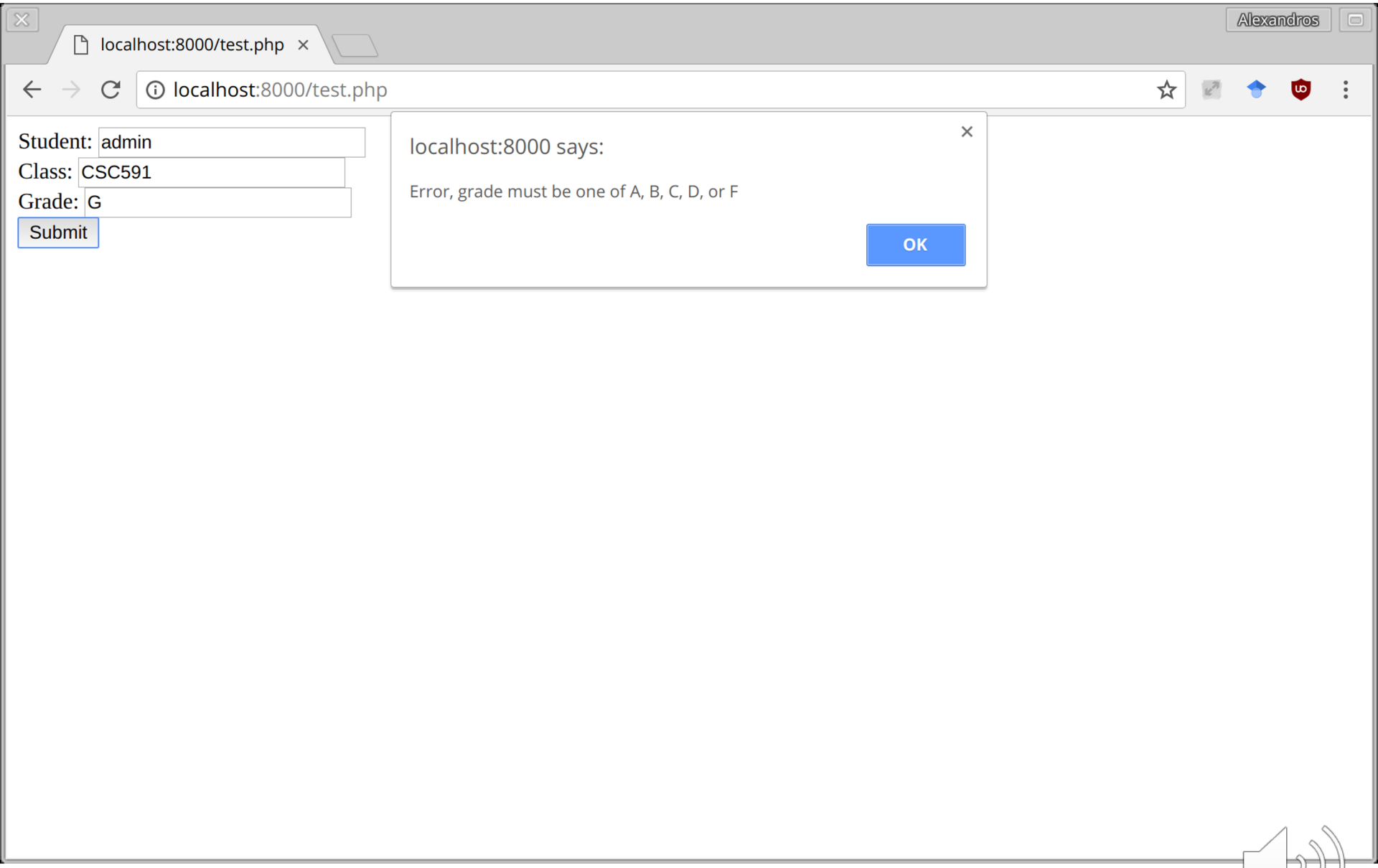
```
<script>
function check_form() {
  var form = document.getElementById("the_form");
  if (form.student.value == "" || form.class.value == "" || form["grade"].value == ""){
            alert("Error, must fill out all the form");
            return false;
  }
  var grade = form["grade"].value;
  if (!(grade == 'A' || grade == 'B' || grade == 'C' ||
        grade == 'D' || grade == 'F')) {
            alert("Error, grade must be one of A, B, C, D, or F");
            return false;
  }
  return true;
}
</script>
<form id="the_form" onsubmit="return check_form()">
 Student: <input type="text" name="student"><br>
 Class:  <input type="text" name="class"><br>
 Grade:  <input type="text" name="grade"><br>
 <input type="submit" name="submit">
</form>
```

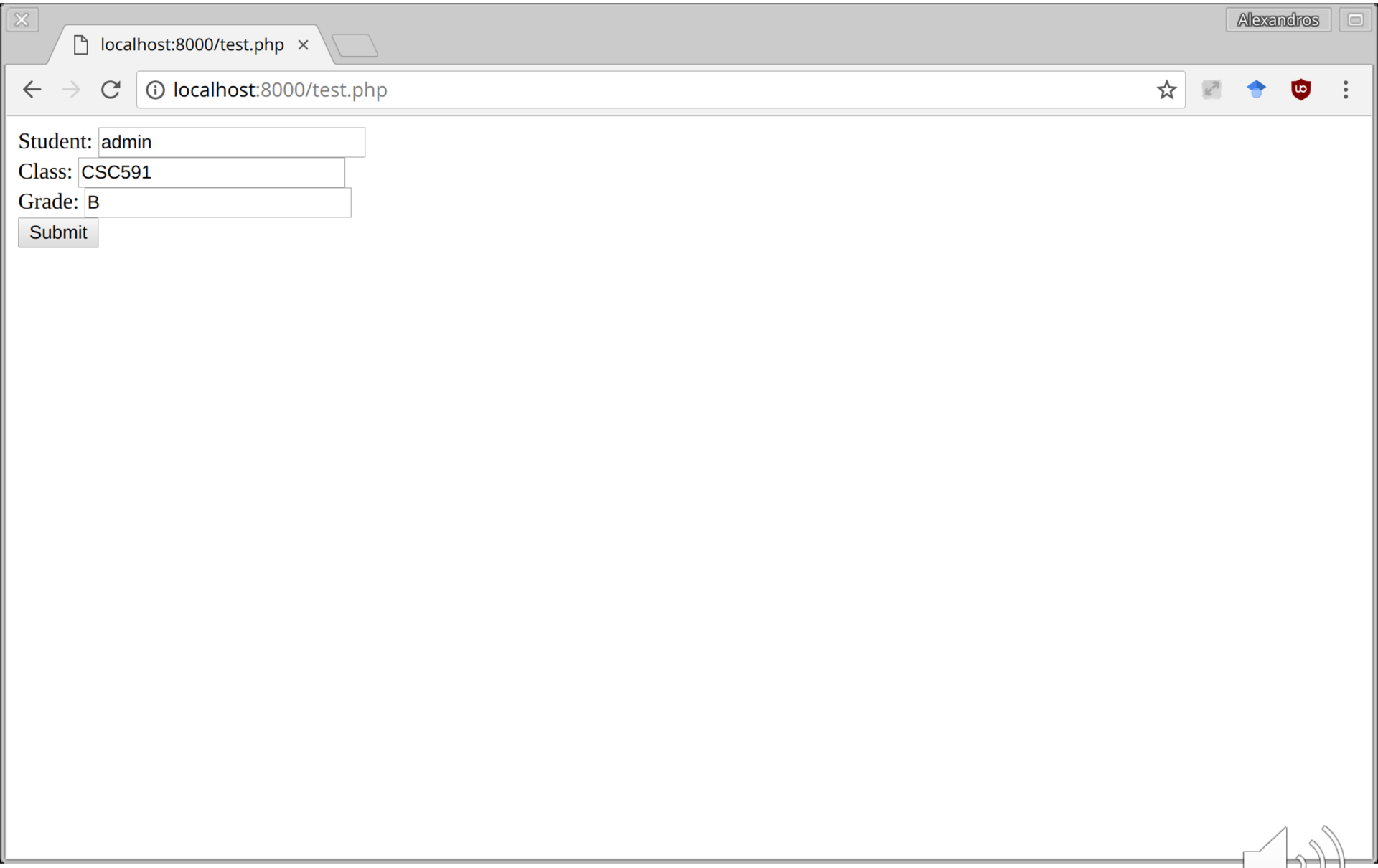

Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student:
Class:
Grade:
Submit

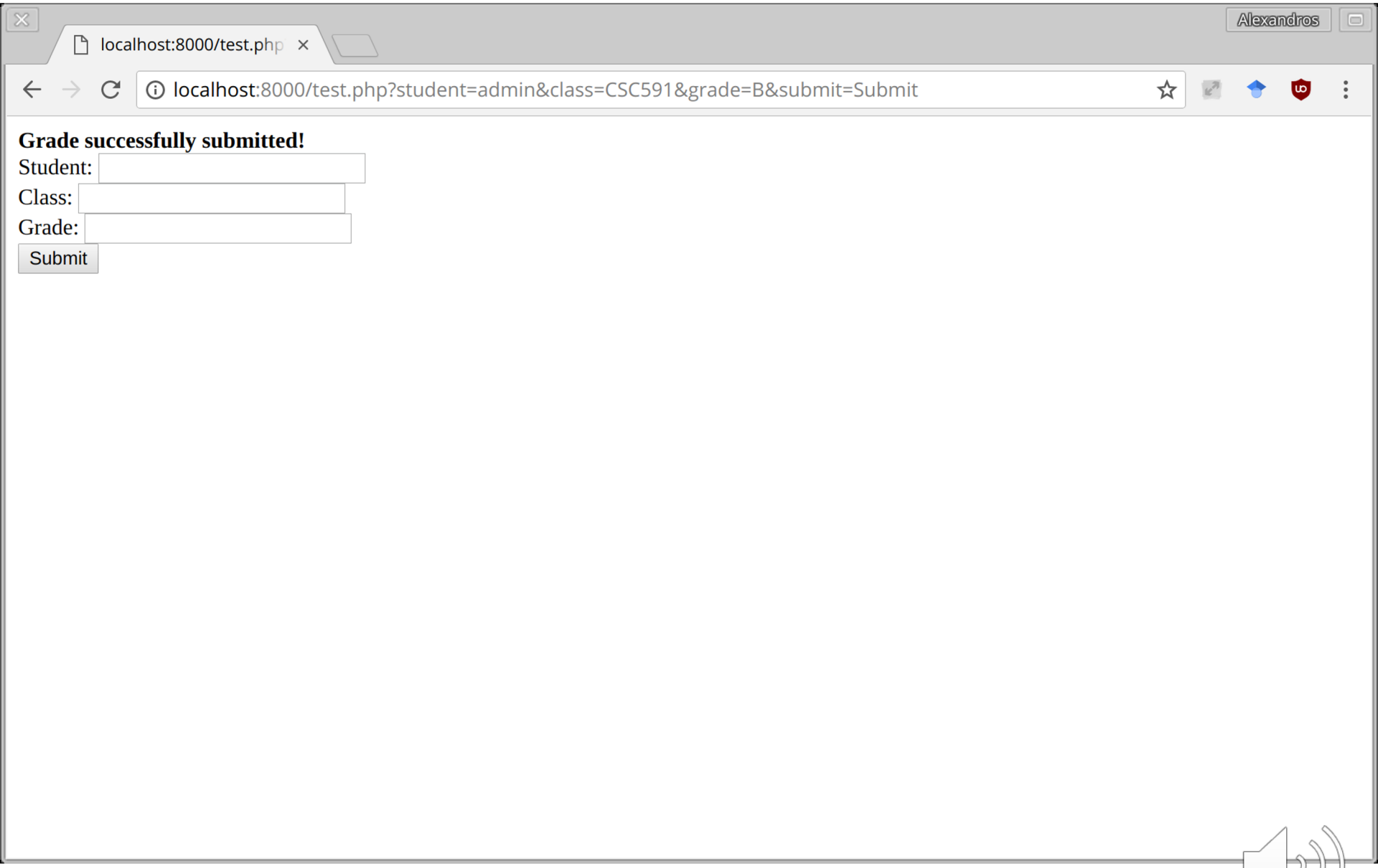

Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class:
Grade: A
Submit

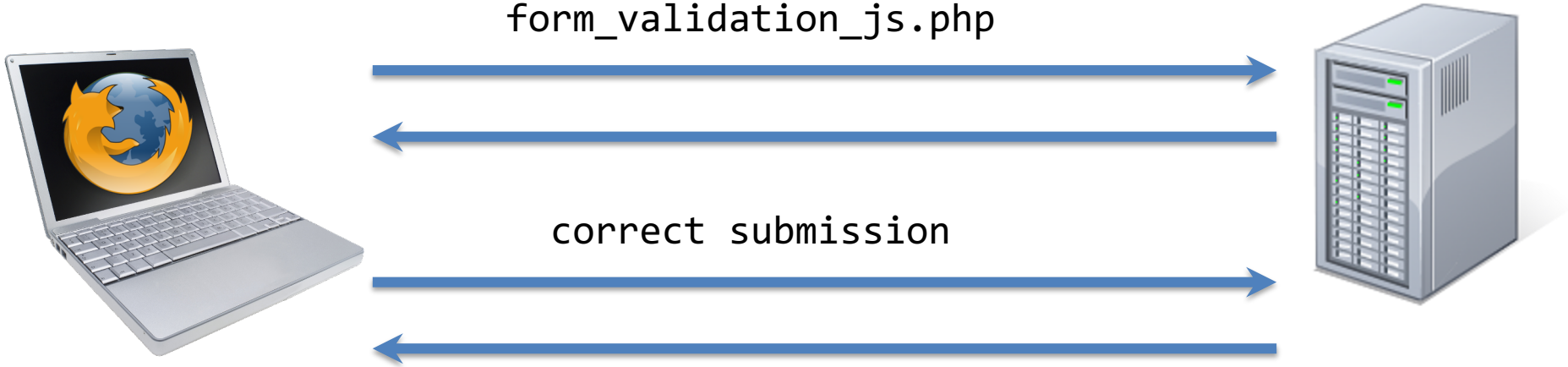

Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class:
Grade: A
Submit
localhost:8000 says:
Error, must fill out all the form
OK


Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class: CSC591
Grade: G
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class: CSC591
Grade: G
Submit
localhost:8000 says:
Error, grade must be one of A, B, C, D, or F
OK


Alexandros
localhost:8000/test.php
localhost:8000/test.php
Student: admin
Class: CSC591
Grade: B
Submit


Alexandros
localhost:8000/test.php
localhost:8000/test.php?student=admin&class=CSC591&grade=B&submit=Submit
Grade successfully submitted!
Student:
Class:
Grade:
Submit


form_validation_js.php
correct submission

Client-Side Validation

- Now that we're doing validation on the client, can we get rid of all those PHP checks in our server-side code?
  - No!
  - No guarantee that client-side validation is performed
    - User disables JavaScript
    - Command-line clients
- Otherwise, users could enter arbitrary data that does not conform to your validation
  - Could lead to a security compromise or not
- So the validation must remain on the server-side and the client-side
  - Brings up another problem, how to perform consistent validation when server-side and client-side written in different languages

source: https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/AJAX_Security_Cheat_Sheet.md