AngularJS and Bootstrap

Remember a MEAN stack?

- MongoDB as the database
- Express as the web framework
- AngularJS as the frontend framework, and
- Node.js as the server platform

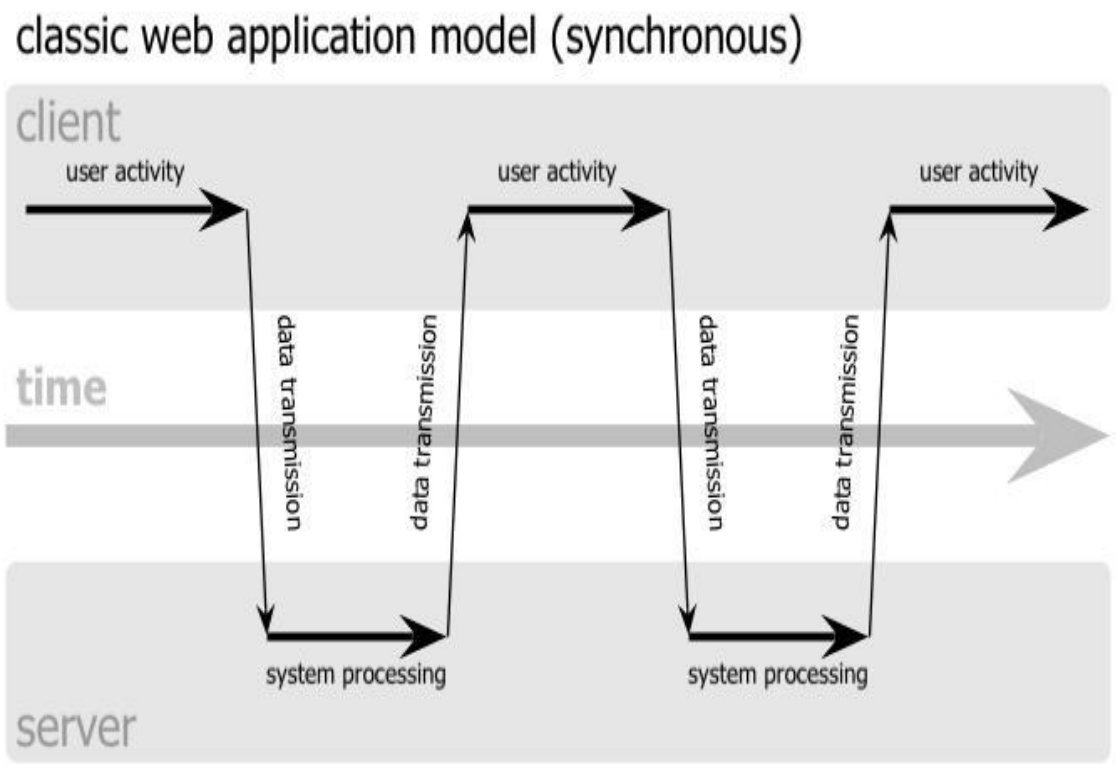
Review JQuery?

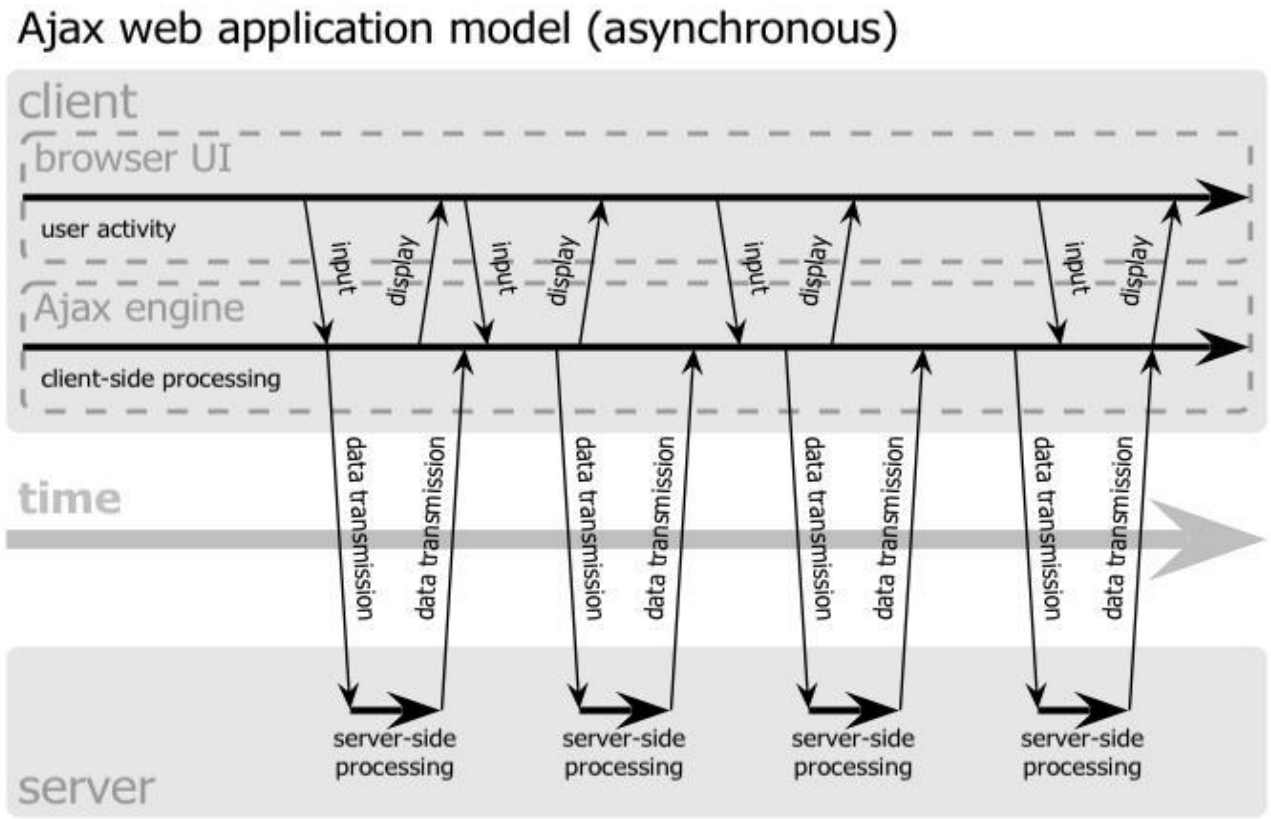
- **JQuery** is a “write less, do more”, JavaScript library
- Purpose of JQuery is to make it much easier to use JavaScript
- Many common tasks in JavaScript are made into functions in JQuery
- Syntax contains selecting HTML elements and performing some action on them
  - $(selector).action()
    - $ sign to define access to JQuery
    - selector to query or find HTML elements
    - action() – jQuery action to be performed
- E.g.: $("button").click();

Review AJAX ?

- **AJAX** – Asynchronous JavaScript and XML
- Main purpose is to exchange data with a server and updating the web page without reloading the whole page
- Allows web pages to update asynchronously by exchanging data with server in the background
- This enables parts of the page to update without a complete page reload
- AJAX uses different technologies for implementation
  - XMLHttpRequest object – to exchange data asynchronously
  - JavaScript – to interact with the information
  - CSS – to style the data
  - XML – used as the format for data transfer

Classic Model vs Ajax Model





What is AngularJS ?

- **AngularJS** is a structural framework for dynamic web apps
- It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components
- Developed focusing on creating single page applications
- Open Source
  - GitHub : https://github.com/angular/angular.js
  - MIT License
- Uses JQuery
- Focuses more on HTML side of the web apps

MVC Architecture

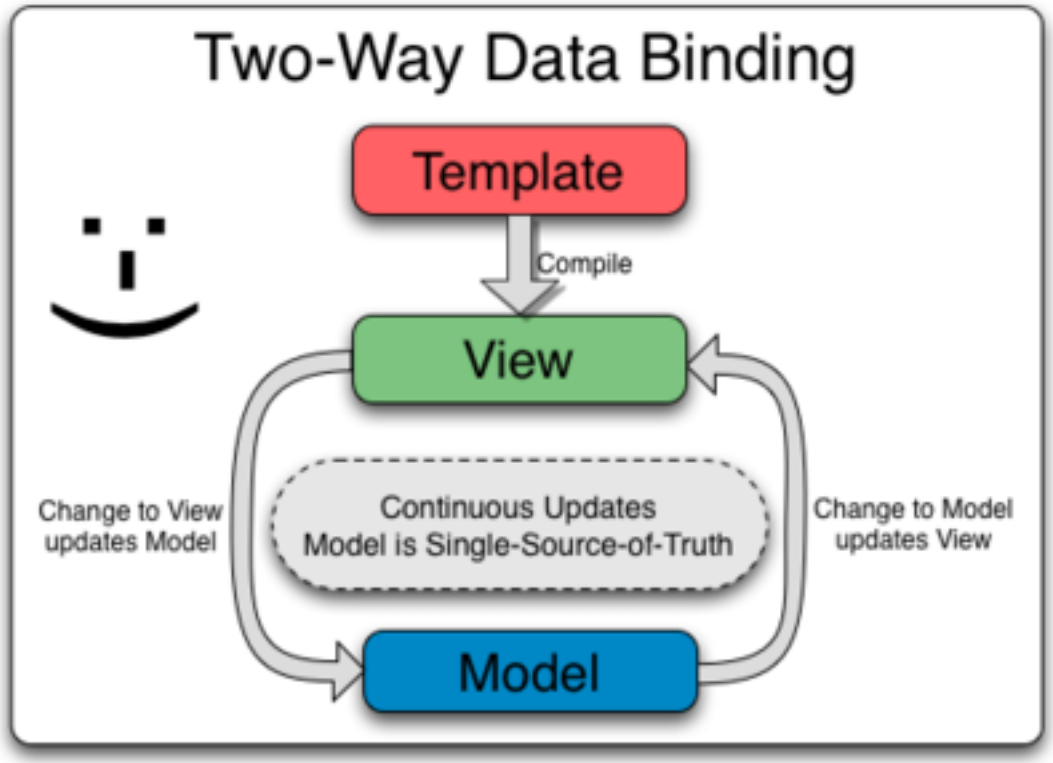
- **M**odel
  - The data
- **C**ontroller
  - The behavior
  - Modifying / updating the models
- **V**iew
  - The interface
  - How the data is presented to the user

JavaScript

HTML

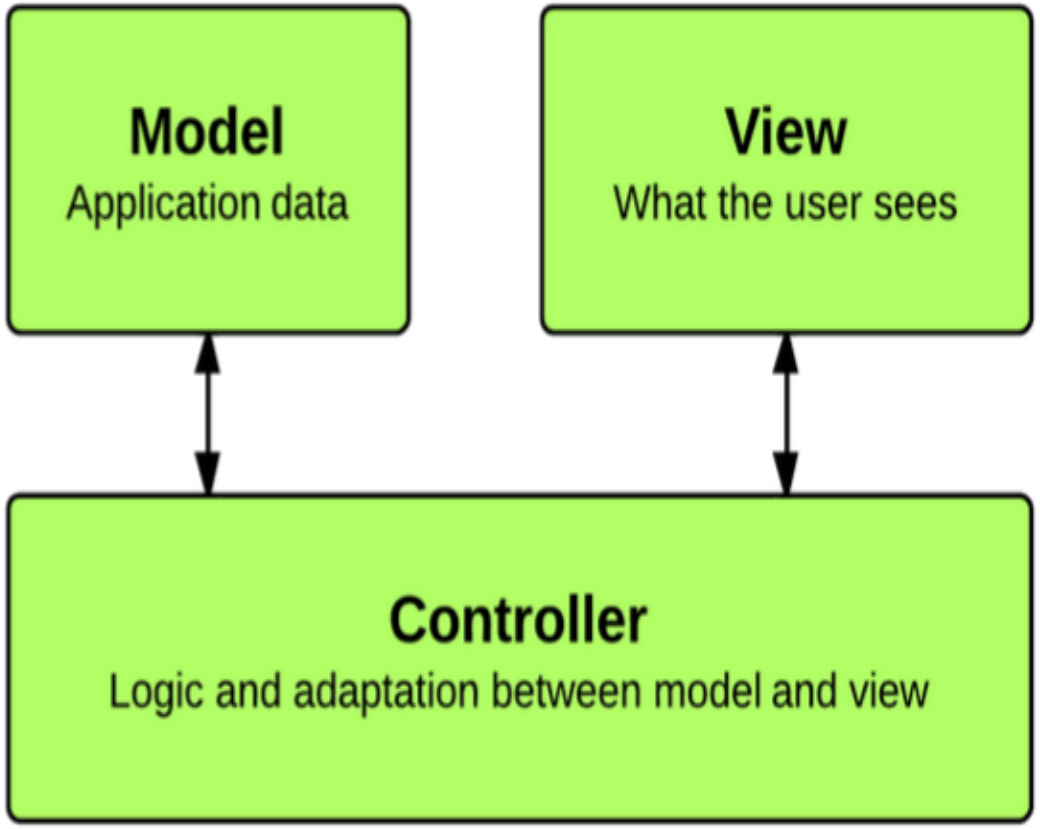
Data Binding

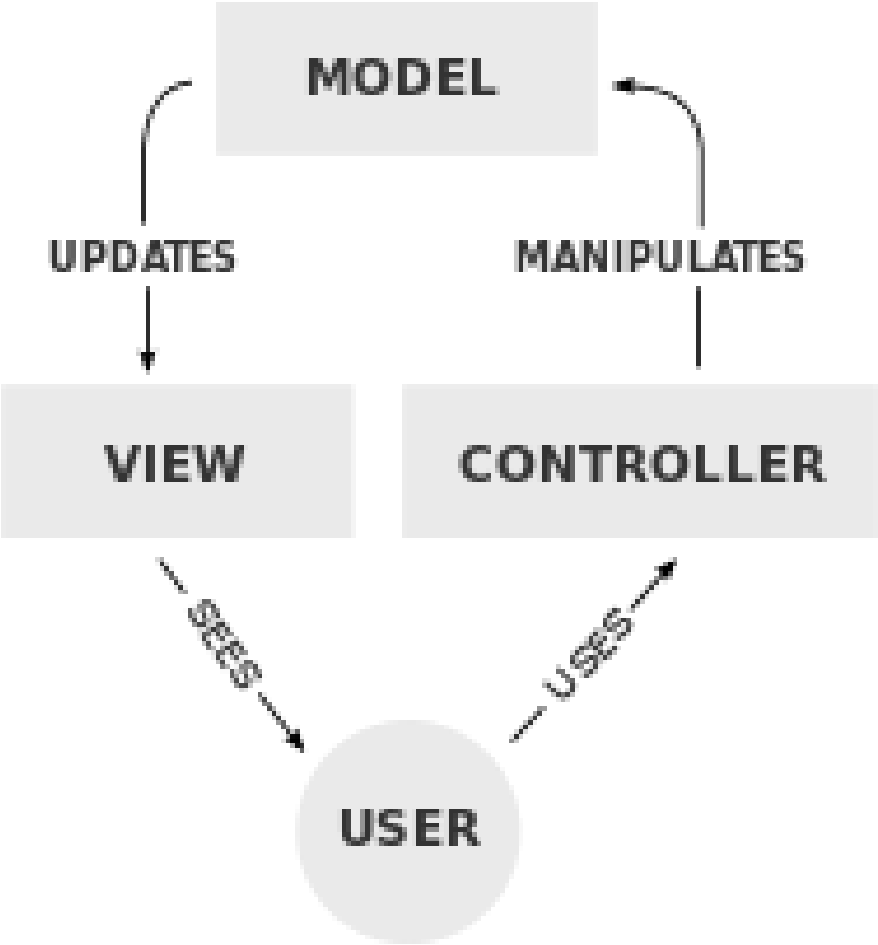
- Views are declarative
  - The data
- Controllers do not directly need to manipulate the view
  - Changes to the model / data is directly reflected in the views
  - Updates are managed by frameworks



https://docs.angularjs.org/guide/databinding

Model View Controller





Model

- Properties on the Controller $scope object
- Standard JavaScript values

Entire model contained in a single javascript data structure.

```
$scope = {
    employeeName: "Mattias",
    company: "Net Insight AB"
}
```

Views

"Extended" html with references to model

```
<h2>{{company}}</h2>
Employee name:
<input ng-model="employeeName">
</input>
<button ng-click="save_info()">Submit</button>
```

Result

Net Insight AB

Employee name: Mattias Submit

http://www.w3schools.com/angular/angular_controllers.asp

Views – AngularJS Directives

- AngularJS directives are extended HTML attributes with the prefix **`ng`**
- **`ng-app`**
  - Initializes the AngularJS application
  - Determines the scope of the AngularJS application
  - If provided a values, it loads the respective module
- **`ng-controller`**
  - Determines the application controller
- **`ng-model`**
  - Binds the value of HTML elements to the application data
  - Provides type validation(number, email, required) or status for application data(invalid, dirty, touched, error)
  - Used for two way binding

http://www.w3schools.com/angular/angular_directives.asp

Views – AngularJS Directives

- **`ng-if`**
  - **`ng-if = “expression” (ng-if = “data.length > 0”)`**
  - Inserts the HTML elements if the expression is true
- **`ng-repeat`**
  - **`ng-repeat = “<variable> in <array>” (<li ng-repeat="x in names">)`**
  - ng-repeat directive is used on array of objects
  - The ng-repeat directive clones HTML elements once for each item in a collection (in an array)

http://www.w3schools.com/angular/ng_ng-if.asp

http://www.w3schools.com/angular/angular_directives.asp

Views – AngularJS Expressions

- AngularJS binds data to HTML using Expressions
- AngularJS expressions are declared inside double braces : **`{{expression}}`**
- AngularJS expressions are much like JavaScript expressions. They can contain liberals, operators, and variables
- Varies from JavaScript expressions in:
  - Evaluated in the current scope(see controllers later).
  - More forgiving to undefined and null errors
  - No control statements conditionals, loop, or throw

http://www.w3schools.com/angular/angular_expressions.asp

Controller

Javascript code that populates the view and reacts to changes in it.

```
function myCtrl( $scope ) {
  $scope = {
     employeeName: "Mattias",
     company: "Net Insight AB"
  };

  $scope.save_info = function() {
        console.log( $scope.employeeName );
  };
}
```

http://www.w3schools.com/angular/angular_controllers.asp

Controller

- Function that takes at least one parameter - $scope
  - The ng-model directives bind the input fields to the controller properties
  - `app.controller('myctrl',function($scope){…});`
- `$scope`
  - JavaScript object
  - Contains data and functions
  - Can add new properties
    - `$scope.<properties_name> = <value>`

Controller

- Dependency Injection
  - Pass the modules and services that you need as parameters
  - In the previous case $scope is a service that will be injected
  - Can be passed as an array of strings to the controller function as well
  - Other useful services:
    - $http
      - Used to handle AJAX calls
      - Wrappers around JQuery

http://www.w3schools.com/angular/angular_modules.asp

Controller

- Typically also contains module loading
- angular.module(<name>,[<dependencies>])
  - Creates a module with the name
  - This module is then configured with controllers

```
var myApp = angular.module('myApp', []);

myApp.controller('MyCtrl', function($scope) { … });
```

Modules

- Modules are used to separate services, controllers and applications
- It keeps the code clean
- Two types of modules:
  - Application Module – used to initialize an application containing controllers
  - Controller Module – used to define the controller
- Application Module
  - `var mainApp = angular.module("mainApp", []);`
  - `<div ng-app = “mainApp”> -` directive mapping to the application module
  - Empty array generally contains the dependency modules
  - Above, the module is named mainApp

Modules

- Controller Module:
  - Contains the module for controllers
  - `<div ng-app="mainApp" ng-controller="mainController">` - directive mapping for the controller
  - `mainApp.controller("mainController", function($scope) {…});`

Twitter Bootstrap

- Popular HTML, CSS and JavaScript framework for developing responsive applications
- Free framework designed to make web application faster and easier
- Includes HTML, CSS based templates for typography, forms, buttons, tables, navigation, images
- Enables application to be more responsive
- Responsive design makes web applications automatically adjust to different devices, resolutions and screens

http://www.w3schools.com/bootstrap/default.asp

Why use Twitter Bootstrap?

State of Today's Web

Why use Twitter Bootstrap ?

- Easy to use:
  - Uses HTML, CSS and JavaScript
  - Anybody with basic knowledge of HTML, CSS can develop Bootstrap applications
- Responsive features:
  - Responsive CSS adjust automatically to phones, tablets, desktops
- Mobile first approach:
  - Mobile first styles part of core bootstrap framework
- Browser compatibility:
  - Compatible with all major browsers – Chrome, Firefox, Internet Explorer, Safari

How to use Twitter Bootstrap ?

- Download Bootstrap:
  - Download and host yourself
  - Get bootstrap from getbootstrap.com
- Install Bootstrap from CDN:
  - Include bootstrap from CDN(Content Delivery Network)

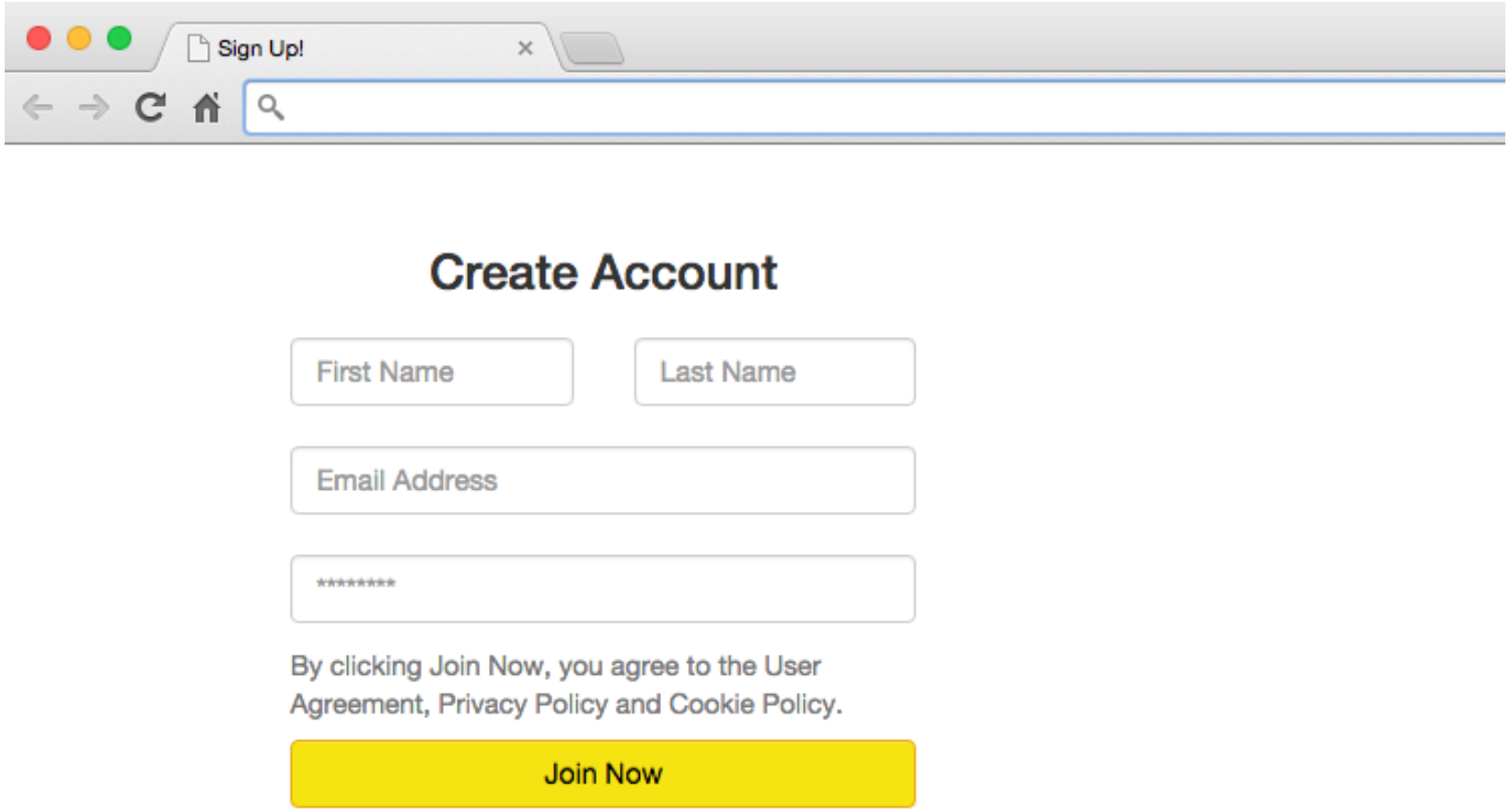
```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.
css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></script>
```

http://www.w3schools.com/bootstrap/bootstrap_get_started.asp

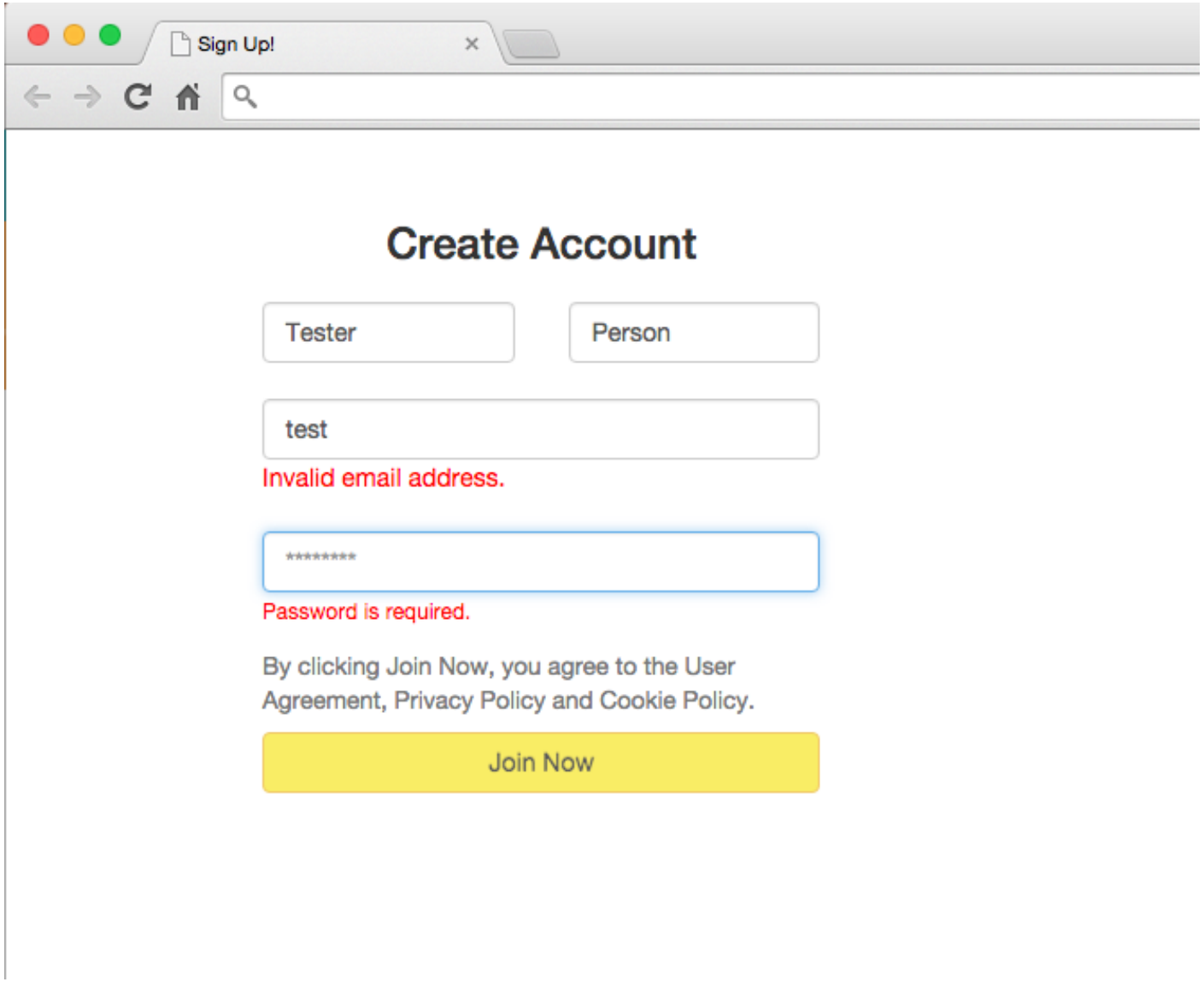
Example 1

- Validate sign up page using AngularJS
- Use bootstrap to design the application

Example 1 – User Interface



Example 1 – AngularJS Validation



Example 1 – Bootstrap Inclusion

```
<div class="row">
    <div class="col-md-12" style="margin-bottom: 10px;">
            <h3 align="center">Create Account</h3>
    </div>
    <div class="col-md-6">
            <input class="form-control" type="text" id="firstname"
                    name="firstname" placeholder="First Name" ng-model="firstname"
                    required>
            <span style="color: red" ng-show="signUpForm.firstname.$dirty && signUpForm.firstname.$invalid"
                    style="font-size: 12px;"> </span>
            <span ng-show="signUpForm.firstname.$error.required"
                    style="font-size: 12px;">First Name is required</span>
    </div>
    <div class="col-md-6">
        <input class="form-control" type="text" id="lastname"
                name="lastname" placeholder="Last Name" ng-model="lastname">
    </div>
</div>
```

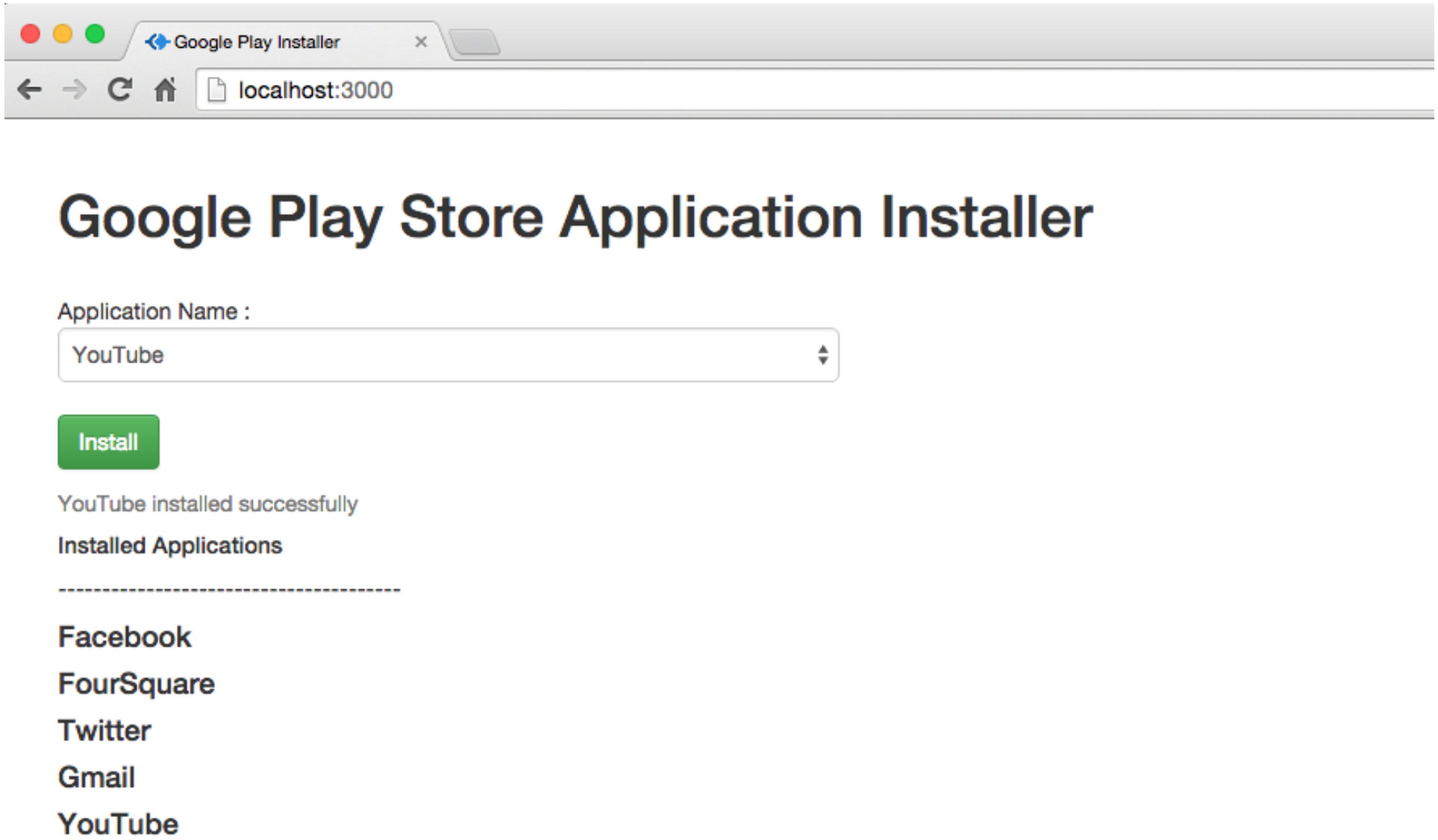
Example 1 – AngularJS

```
<input class="form-control" type="email" name="user" ng-model="user" id="registerEmail"
name="registerEmail" placeholder="Email Address" style="margin-top: 20px;" required>

<span style="color: red" ng-show="signUpForm.user.$dirty && signUpForm.user.$invalid">
    <span ng-show="signUpForm.user.$error.required">Email is required.</span>
    <span ng-show="signUpForm.user.$error.email">Invalid email address.</span>
</span>

<input class="form-control" type="password" id="registerPassword" name="registerPassword" style="margin-top: 20px;" placeholder="********"
ng-model="registerPassword" required>

<span style="color: red" ng-show="signUpForm.registerPassword.$dirty &&
signUpForm.registerPassword.$invalid" style="font-size: 12px;">
    <span ng-show="signUpForm.registerPassword.$error.required" style="font-size:
12px;">Password is  required.</span>
</span>
```

Example 2

- Google Play Installer
- Install application, if it is present on google play install,
- If already installed, give message
- If not installed, give message “Application Installed”

Example 2 – User Interface



Example 2 – Bootstrap Inclusion

```
<div class="row" >

        <div class="col-md-12">
                <h1>Google Play Store Application Installer</h1>
        </div>

        <div class="col-md-12" style="margin-bottom: 20px;">
                Application Name :
                //Form
                <form method="post" ng-submit="submit();" method="post" name="google_play_selector" novalidate>
                        <select id="applicationInstall" name="applicationInstall" ng-model="applicationInstall"
class="form-control" style="width: 40%;">
                                <option value="InstallApplication">Install Application</option>
                                <option value="Facebook">Facebook</option>
                        </select> <br>
                        <input type="submit" value="Install" class="btn btn-success" />
                </form>

                <h4 id="message"><small>{{message}}</small></h4>
                <h5>Installed Applications</h5>
                ---------------------------------------
                <div ng-repeat="app in applicationsInstalled">
                        <h4>{{app}}</h4>
                </div>

        </div>
</div>
```

Example 2 – AngularJS directives

```
<body ng-app="google_play" ng-controller="google_play_controller">
<div class="row" >
        <div class="col-md-12">
            <h1>Google Play Store Application Installer</h1>
        </div>
        <div class="col-md-12" style="margin-bottom: 20px;">
            Application Name :
            //Form
            <form method="post" ng-submit="submit();" method="post" name="google_play_selector" novalidate>
                <select id="applicationInstall" name="applicationInstall"
                ng-model="applicationInstall">
                    <option value="InstallApplication">Install Application</option>
                    <option value="Facebook">Facebook</option>
                </select> <br>
                <input type="submit" value="Install" class="btn btn-success" />
            </form>

            <h4><small>{{message}}</small></h4>
            <h5>Installed Applications</h5>
            ----------------------------------------
            <div ng-repeat="app in applicationsInstalled">
                <h4>{{app}}</h4>
            </div>
        </div>
</div>
</body>
```

Example 2 – AngularJS controller

```
<script>
        var google_play = angular.module('google_play', []);                    //Modules – Google Play module
        google_play.controller('google_play_controller',function($scope, $http) { // Google Play controller

                $scope.submit = function() {
                        $http({
                                method : "POST",
                                url : '/installApplication',
                                data : {"applicationInstall" : $scope.applicationInstall}
                                })
                                .success(function(data) {
                                        $scope.message = data.message;
                                        $scope.applicationsInstalled = data.applicationsInstalled;

                                })
                                .error(function(error) {
                                        alert("error");
                                });
                };
    });
    </script>
```

Exercise

- Create a Hotel order system using AngularJS and Bootstrap
- It will be a single page application
- Create a section for menu list containing items and its price displayed and select button for each item
- When you submit the order, a receipt for the order should be generated with the total amount and the number of items on the receipt
- The receipt should display each item with its price and at the bottom of the receipt, you should display the total amount for the receipt.
- Use bootstrap to make your application User interface look good

References

- AngularJS - angularjs.org
- AngularJS W3Schools - w3schools.com/angular
- Bootstrap - getbootstrap.com
- Bootstrap W3Schools - w3schools.com/bootstrap