

# Lab: Creating PizzaMore Web Site using Java, Bootstrap and CGI

This **tutorial** provides step-by-step guidelines to build a “**PizzaMore**” app in Java, HTML, CSS, Bootstrap and CGI. The app should implement **sign in / sing up / main / menu** pages.

## Project Specification

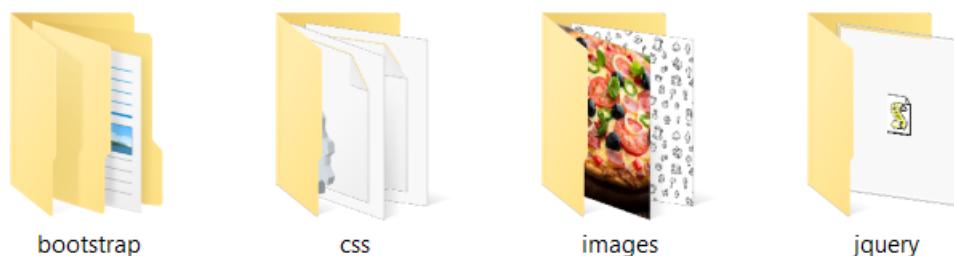
Design and implement a “**PizzaMore**” **web application** in Java, HTML, CSS, Bootstrap and CGI. Create 4 HTML pages with the following functionality:

- **Home**
  - o Home page of the pizza site
  - o Should be able to redirect to all other pages
  - o Should send a **cookie** to the browser
  - o Option to change language via **cookie**
- **Sign Up**
  - o Register a new user in a the database.
- **Sign In**
  - o In case of success, a session should be stored in your database.
  - o Log out option that will delete the session

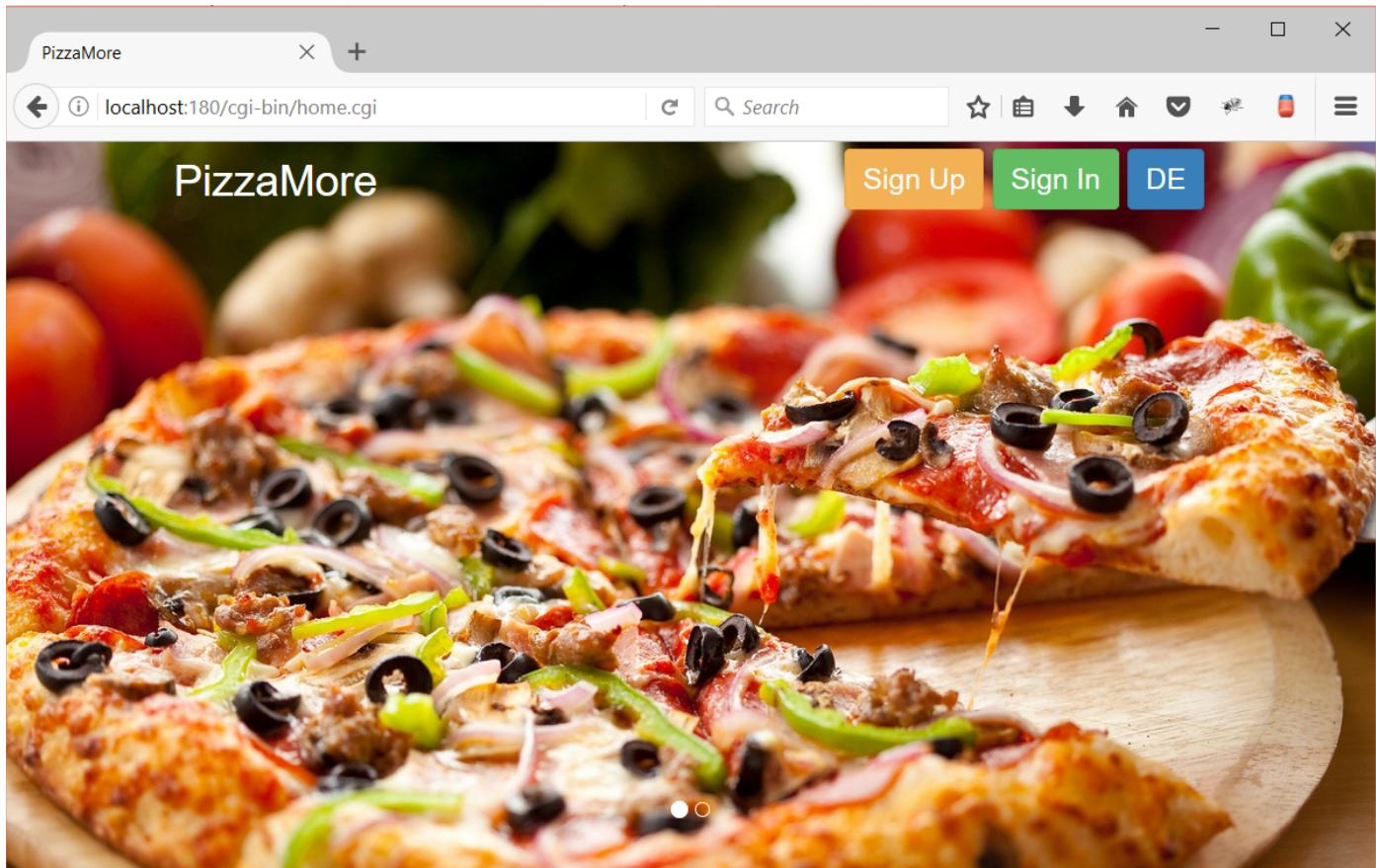
## 1. Setup

- Download [Bootstrap](#) and place it in **htdocs folder**.
- Download [jQuery](#) and place it in **htdocs folder**.
- Create a **css folder** in **htdocs folder**.
- Create an **image folder** in **htdocs folder**. **Copy the images from the archive.**

↑  > This PC > Local Disk (C:) > Apache24 > htdocs >



## 2. Build Home Web Page



Create an empty HTML page **home.html**.

- The design includes two Bootstrap features:
  - Carousel
  - Navigation
- Put the required metadata in the **<head>** tag. Bootstrap link should be located there.

```
home.html

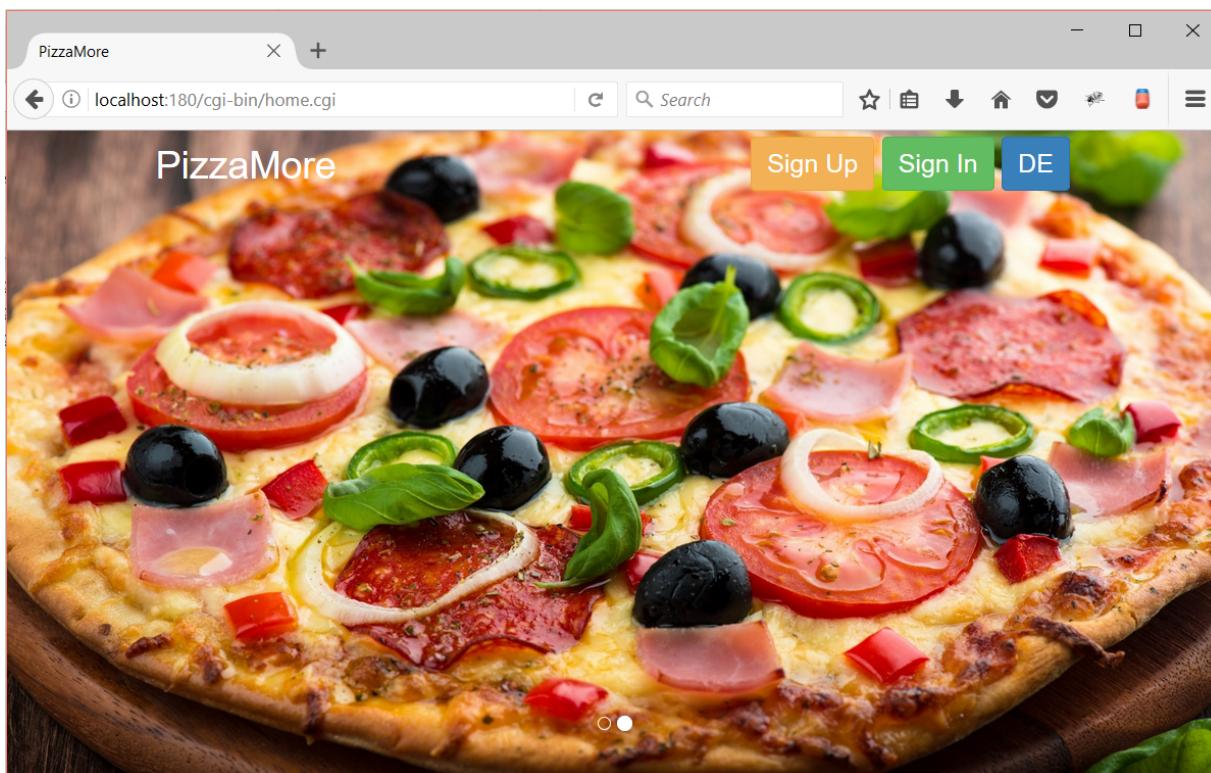
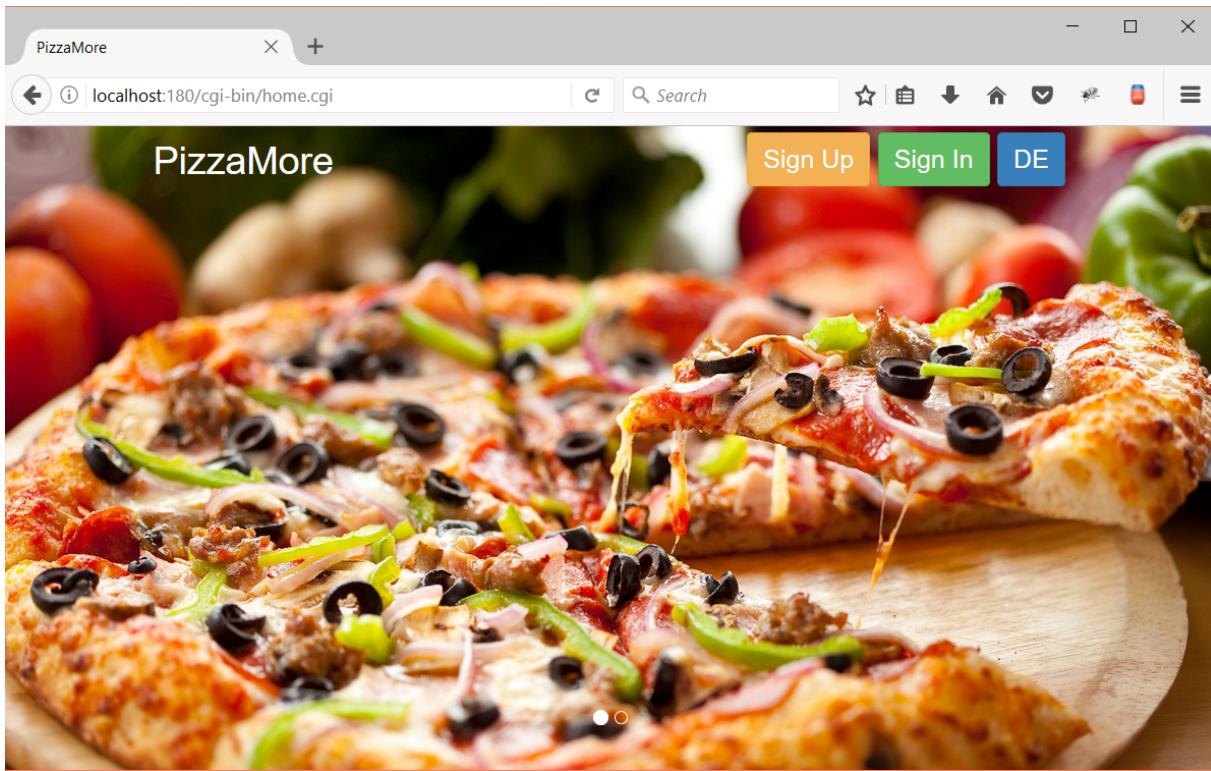
<head>
  <title>PizzaMore</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" type="text/css" href="/bootstrap/css/bootstrap.min.css"/>
  <link rel="stylesheet" type="text/css" href="/css/main.css"/>
</head>
```

- Import the required JavaScript module and jQuery just above the closing **</body>** tag.

```
home.html

<script src="/jquery/jquery.min.js"></script>
<script src="/bootstrap/js/bootstrap.min.js"></script>
</body>
```

- Create a new Carousel. Its main functionality will be to **slide** between pictures.

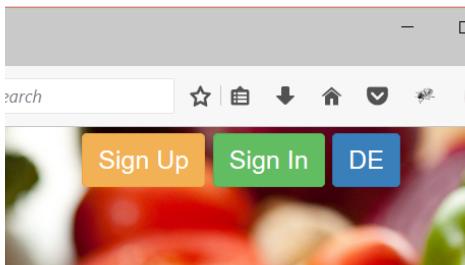


## home.html

```
<div class="container-fluid">
  <div id="myCarousel" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
      <li data-target="#myCarousel" data-slide-to="1"></li>
    </ol>

    <div class="carousel-inner" role="listbox">
      <div class="item active">
        
      </div>
      <div class="item">
        
      </div>
    </div>
  </div>
</div>
```

- Create a navigation with couple with 4 buttons
  - o Sign Up should lead to the sign up section: **signup.cgi**
  - o Sign In should lead to the sign in section: **signin.cgi**
  - o EN should lead to the home section: **home.cgi** and it should change the languages of the page



## home.html

```
<nav id="navigation" class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">PizzaMore</a>
    </div>
    <form method="post">
      <input class="btn btn-primary navbar-btn" type="submit" name="language" value="DE"/>
      <input class="btn btn-success navbar-btn" type="submit" name="signin" value="Sign In"/>
      <input class="btn btn-warning navbar-btn" type="submit" name="signup" value="Sign Up"/>
    </form>
  </div>
</nav>
```

- Create a file called **main.css** and put it in **htdocs/css**. The main styling of the **home** page should be there
- Make the pictures take the whole screen

```
main.css

#myCarousel {
    height: 100%;
    width: auto;
    position: absolute;
}

.carousel-inner {
    height: 100%;
    width: auto;
}

.container-fluid {
    padding: 0;
}
```

- Format the navigation to be **on top of the Carousel**.

```
main.css

#navigation {
    background-color: rgba(255,0,0,0);
    border: none;
    position: relative;
    padding: 2px;
    font-size: 20px;
}

.btn {
    font-size: 20px;
    float: right;
    margin: 3px;
}
```

- Format the **default bootstrap settings**. Copy them from the box below.

```
main.css

.navbar-default .navbar-brand {
    color: #ffffff;
}
.navbar-default .navbar-brand:hover,
.navbar-default .navbar-brand:focus {
    color: #e0e0e0;
    font-size: 30px;
}
```

- Create a **Java** class for the cookies. It should have a method that returns the name and the value with “=” between them.

## Cookie.java

```
public class Cookie {  
    private String name;  
  
    private String value;  
  
    public Cookie(String name, String value) {  
        this.setName(name);  
        this.setValue(value);  
    }  
  
    //TODO Getters and Setters  
  
    public String getCookieValue() {  
        return this.name + "=" + this.value + "; ";  
    }  
}
```

- Create a **Java** file Header.java. It should take care of some of the possible headers

## Header.java

```
public class Header {  
  
    private String type;  
  
    private String location;  
  
    private List<Cookie> cookies;  
  
    public Header() {  
        this.type = "Content-type: text/html;";  
        this.cookies = new ArrayList<>();  
    }  
  
    public void addLocation(String location) {  
        this.location = location;  
    }  
  
    public void addCookie(Cookie cookie) {  
        this.cookies.add(cookie);  
    }  
  
    public void printHeader() {  
        System.out.println(this.type);  
  
        if (!this.cookies.isEmpty()) {  
            String cookies = "";  
            for (Cookie cookie : this.cookies) {  
                cookies += cookie.getCookieValue();  
            }  
  
            System.out.println("Set-Cookie: " + cookies);  
        }  
  
        if (this.location != null) {  
            System.out.println("Location: " + this.location);  
        }  
  
        //End of header  
        System.out.println();  
    }  
}
```

- Create a **Java** file Home.java. It should take care of the actions in the Home page

## Home.java

```
public class Home {  
  
    private static Map<String, String> parameters;  
  
    private static Map<String, Cookie> cookies;  
  
    private static Header header;  
  
    static {  
        parameters = new HashMap<>();  
        cookies = new HashMap();  
        header = new Header();  
    }  
}
```

- Then you need to read the **incoming parameters and take some actions**. Use the [WebUtils](#) class.

## Home.java

```
public static void readParameters() {  
    parameters = WebUtils.getParameters();  
    for (String param : parameters.keySet()) {  
        switch (param) {  
            case "language":  
                String value = parameters.get("language");  
                setCookie("lang", value);  
                break;  
            case "signin":  
                goToSignIn();  
                break;  
            case "signup":  
                goToSignUp();  
                break;  
        }  
    }  
}  
  
private static void goToSignUp() {  
    header.addLocation("http://localhost:180/cgi-bin/signup.cgi");  
}  
  
private static void goToSignIn() {  
    header.addLocation("http://localhost:180/cgi-bin/signin.cgi");  
}  
  
private static void setCookie(String key, String value) {  
    Cookie cookie = new Cookie(key, value);  
    header.addCookie(cookie);  
}
```

- Then we need to read the **incoming cookies**:

## Home.java

```
private static void readCookies(String[] args) {
    if (args.length == 0) {
        return;
    }

    for (String incomingCookie : args) {
        String[] tokens = incomingCookie.split("=");
        String key = tokens[0];
        String value = tokens[1];
        value = value.replace(";", "");
        Cookie cookie = new Cookie(key, value);
        cookies.put(key, cookie);
    }
}
```

- The last step would be to **print the html**

## Home.java

```
private static void readHtml() {

    Cookie languageCookie;
    if (!WebUtils.isPost()) {
        if (cookies.containsKey("lang")) {
            languageCookie = cookies.get("lang");
            if (languageCookie.getValue().equals("DE")) {
                readHtmlDe();
            } else {
                readHtmlEn();
            }
        }
    } else {
        if (parameters.containsKey("language")) {
            String language = parameters.get("language");
            if (language.equals("DE")) {
                readHtmlDe();
            } else {
                readHtmlEn();
            }
        }
    }
}

private static void readHtmlEn(){
}

private static void readHtmlDe(){
}
```

- Finally run the main method

## Home.java

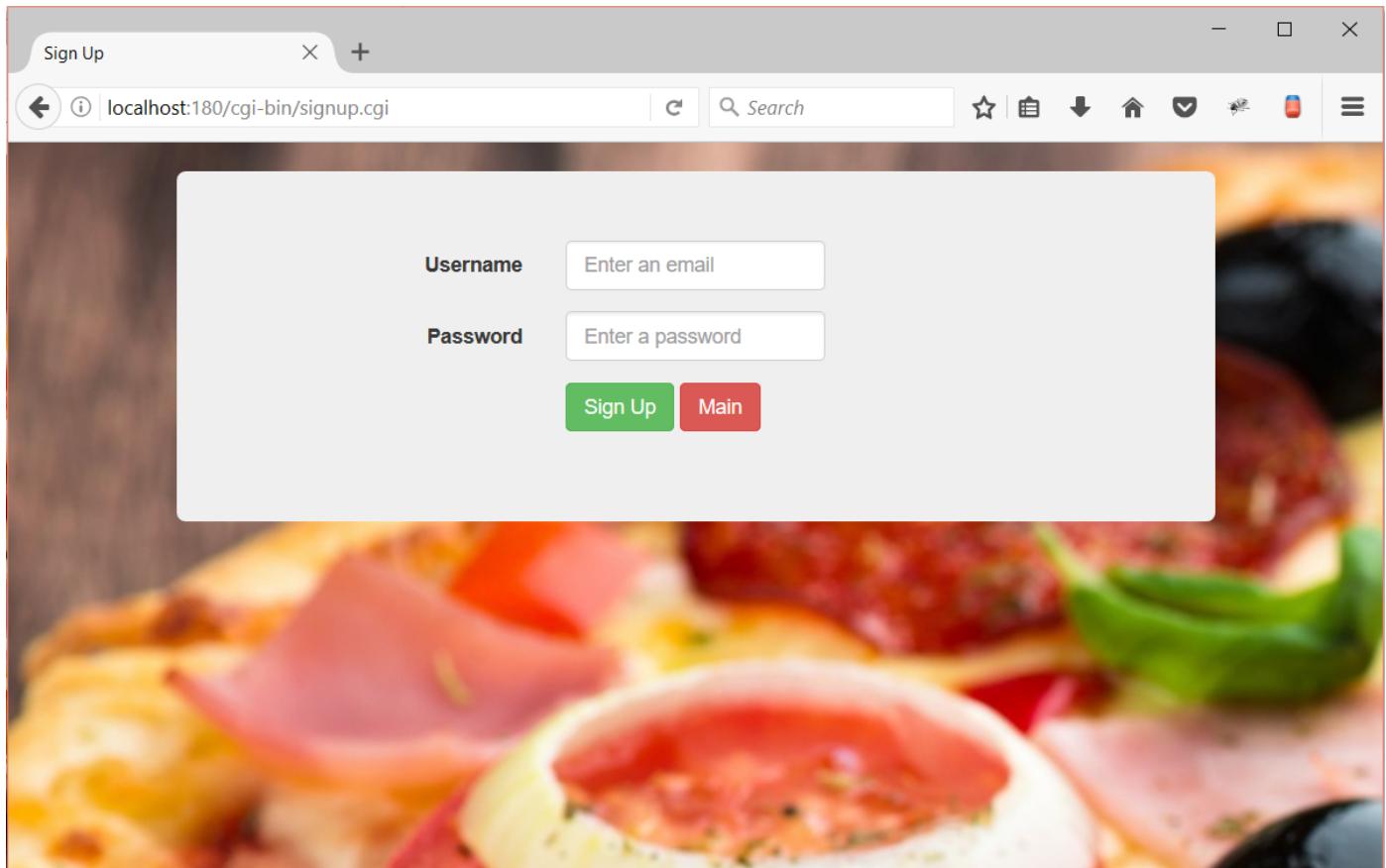
```
public static void main(String[] args) {
    readParameters();
    header.printHeader();
    readCookies(args);
    readHtml();
}
```

- Create a CGI file

### home.cgi

```
#!C:\cygwin64\bin\bash.exe
java -Dcgi.query_string=$QUERY_STRING -Dcgi.request_method=$REQUEST_METHOD -jar Home
$HTTP_COOKIE
```

## 3. Build Sign Up Web Page



Create an empty HTML page **signup.html**.

- The design includes two Bootstrap features:
  - Form
  - Jumbotron
- Create the form using **Bootstrap**. Don't forget to import the required **links**.

## signup.html

```
<br/>
<div class="container">
    <div class="jumbotron">
        <form method="post" class="form-horizontal">
            <div class="form-group">
                <label class="control-label col-sm-2 col-sm-offset-2">Username</label>
                <div class="col-sm-4">
                    <input class="form-control" name="username" type="email"
placeholder="Enter an email"/>
                </div>
            </div>
            <div class="form-group">
                <label class="control-label col-sm-2 col-sm-offset-2">Password</label>
                <div class="col-sm-4">
                    <input class="form-control" name="password" type="password"
placeholder="Enter a password"/>
                </div>
            </div>
            <div class="form-group">
                <div class="col-sm-offset-4 col-sm-4">
                    <input class="btn btn-success" type="submit" name="signup"
value="Sign Up" />
                    <input class="btn btn-danger" type="submit" name="main" value="Main"
/>
                </div>
            </div>
        </form>
    </div>
</div>
```

- Add the background image via css:

## signup.css

```
body {
    background-image: url("/images/pizza_2.jpg");
}
```

- Create **User** class

## User.java

```
@Entity
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    private String name;

    private String password;

    public User() {
    }

    public User(String name, String password) {
        this.setName(name);
        this.setPassword(password);
    }
}
```

- Create user repository that will retrieve and save users

## UserRepository.java

```
public class UserRepository {

    public void createUser(User user) {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("pizzaMore");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();
        em.persist(user);
        em.getTransaction().commit();
        em.close();
        emf.close();
    }

    public User findByUserAndPassword(String username, String password) {
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("pizzaMore");
        EntityManager em = emf.createEntityManager();
        em.getTransaction().begin();
        Query query = em.createQuery("SELECT u FROM User AS u " +
                "WHERE u.name = :username " +
                "AND u.password = :password");
        query.setParameter("username", username);
        query.setParameter("password", password);
        User user = null;
        if(!(query.getResultList().size() == 0)){
            user = (User) query.getSingleResult();
        }

        em.getTransaction().commit();
        em.close();
        emf.close();
        return user;
    }
}
```

- Create SignUp java class that will **save user's details into a database**.

### SignUp.java

```
public class SignUp {

    private static Map<String, String> parameters;

    private static Header header;

    private static UserRepository userRepository;

    static {
        parameters = new HashMap<>();
        header = new Header();
        userRepository = new UserRepository();
    }
}
```

- Create a method that will read the parameters

### SignUp.java

```
public static void readParameters() {
    parameters = WebUtils.getParameters();
    String username = null;
    String password = null;
    for (String param : parameters.keySet()) {
        switch (param) {
            case "username":
                username = parameters.get("username");
                break;
            case "password":
                password = parameters.get("password");
                break;
            case "signup":
                if (username.isEmpty() || password.isEmpty()) {
                    return;
                }

                User user = new User(username, password);
                createUser(user);
                header.addLocation("http://localhost:180/cgi-bin/signin.cgi");
                break;
            case "main":
                header.addLocation("http://localhost:180/cgi-bin/home.cgi");
                break;
        }
    }
}
```

- Create a method that will create the user

### SignUp.java

```
public static void createUser(User user) {
    userRepository.createUser(user);
}
```

- Invoke the main method

```

SignUp.java

public static void main(String[] args) {
    readParameters();
    header.printHeader();
    readHtml();
}
```

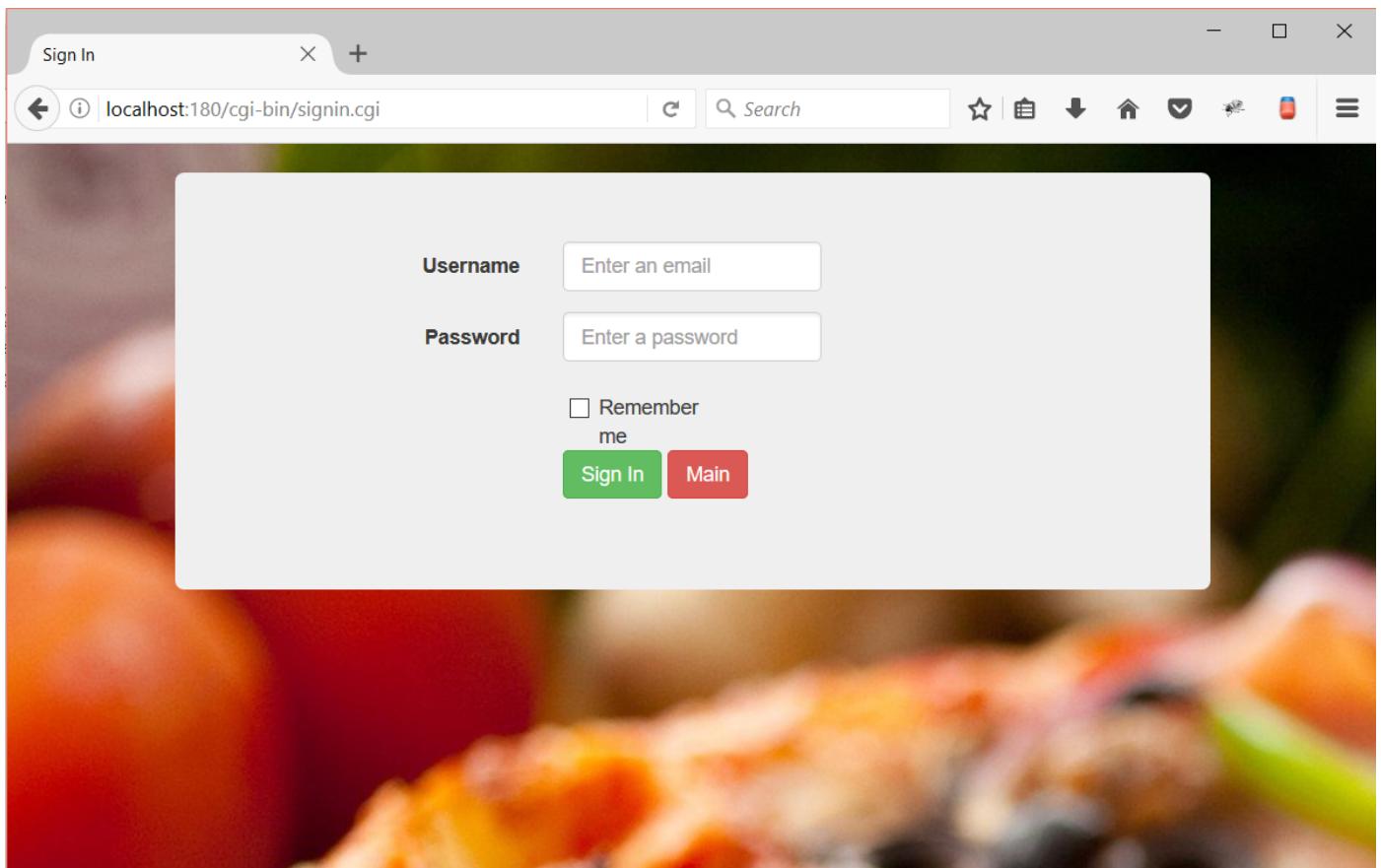
- Create a CGI file

```

singup.cgi

#!/cygwin64/bin/bash.exe
java -Dcgi.query_string=$QUERY_STRING -Dcgi.request_method=$REQUEST_METHOD -jar
SignUp.jar
```

## 4. Build Sign In Web Page



Create an empty HTML page **signin.html**.

- The design includes two Bootstrap features:
  - o Form
  - o Jumbotron
- Create the form using **Bootstrap**. Don't forget to import the required **links**.

## singin.html

```
<br/>
<div class="container">
    <div class="jumbotron">
        <form method="post" class="form-horizontal">
            <div class="form-group">
                <label class="control-label col-sm-2 col-sm-offset-2">Username</label>
                <div class="col-sm-4">
                    <input class="form-control" name="username" type="email"
placeholder="Enter an email"/>
                </div>
            </div>
            <div class="form-group">
                <label class="control-label col-sm-2 col-sm-offset-2">Password</label>
                <div class="col-sm-4">
                    <input class="form-control" name="password" type="password"
placeholder="Enter a password"/>
                </div>
            </div>
            <div class="checkbox col-sm-2 col-sm-offset-4">
                <label>
                    <input type="checkbox" name="rememberme">
                    <span>Remember me</span>
                </label>
            </div>
            <br/>
            <br/>
            <div class="form-group">
                <div class="col-sm-offset-4 col-sm-4">
                    <input class="btn btn-success" type="submit" name="signin"
value="Sign In" />
                    <input class="btn btn-danger" type="submit" name="main" value="Main"
/>
                </div>
            </div>
        </form>
    </div>
</div>
```

- Add a **background image via css** file:

## signin.css

```
body {
    background-image: url("/images/pizza_1.jpg");
}
```

- Create a java file called **SignIn.java**
  - **It should check the credentials**
  - **Create a session with them**
  - **Send the session as a cookie**
  - **When signed out the session should be deleted**
- Create a Session class

## Session.java

```
@Entity
@Table(name = "sessions")
public class Session {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @OneToMany(mappedBy = "session", cascade = CascadeType.ALL, fetch = FetchType.EAGER,
    orphanRemoval=true)
    private Set<SessionData> sessionData;

    public Session() {
        this.setSessionData(new HashSet<>());
    }

    public void addData(String key, String value){
        this.getSessionData().add(new SessionData(key,value, this));
    }
}

@Entity
@Table(name = "sessions_data")
public class SessionData {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @Column(name = "key_data")
    private String key;

    @Column(name = "value_data")
    private String value;

    @ManyToOne
    @JoinColumn(name = "session_id")
    private Session session;

    public SessionData() {
    }

    public SessionData(String key, String value, Session session) {
        this.setKey(key);
        this.setValue(value);
        this.setSession(session);
    }
}
```

- o Create Session Repository which will save and retrieve sessions

## SessionRepository.java

```
public class SessionRepository {  
  
    public long createSession(Session session){  
        long id;  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("pizzaMore");  
        EntityManager em = emf.createEntityManager();  
        em.getTransaction().begin();  
        em.persist(session);  
        id = session.getId();  
        em.getTransaction().commit();  
        em.close();  
        emf.close();  
        return id;  
    }  
  
    public Session findById(long id){  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("pizzaMore");  
        EntityManager em = emf.createEntityManager();  
        em.getTransaction().begin();  
        Query query = em.createQuery("SELECT s FROM Session AS s " +  
            "WHERE s.id = :id");  
        query.setParameter("id", id);  
        Session session = null;  
        if(!(query.getResultList().size() == 0)){  
            session = (Session) query.getSingleResult();  
        }  
  
        em.getTransaction().commit();  
        em.close();  
        emf.close();  
        return session;  
    }  
  
    public void delete(Long id) {  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory("pizzaMore");  
        EntityManager em = emf.createEntityManager();  
        em.getTransaction().begin();  
        Query query = em.createQuery("DELETE FROM Session AS s " +  
            "WHERE s.id = :id");  
        query.setParameter("id", id);  
        query.executeUpdate();  
        em.getTransaction().commit();  
        em.close();  
        emf.close();  
    }  
}
```

- Create a method a **SingIn** java class

### SignIn.java

```
private static Map<String, String> parameters;  
  
private static Map<String, Cookie> cookies;  
  
private static Header header;  
  
private static UserRepository userRepository;  
  
private static SessionRepository sessionRepository;  
  
static {  
    parameters = new HashMap<>();  
    cookies = new HashMap();  
    header = new Header();  
    userRepository = new UserRepository();  
    sessionRepository = new SessionRepository();  
}  
}
```

- Create a method that will read the **parameters** and take the necessary actions. **Send a session cookie with the session id to the client.**

## SignIn.java

```
public static void readParameters() {
    parameters = WebUtils.getParameters();
    for (String param : parameters.keySet()) {
        switch (param) {
            case "signin":
                signIn();
                break;
            case "main":
                goToMain();
                break;
        }
    }
}

private static void signIn() {
    String username = parameters.get("username");
    String password = parameters.get("password");
    User user = userRepository.findByUserAndPassword(username,
password);
    if(user != null) {
        Session session = new Session();
        session.addData("username", username);
        long sid = sessionRepository.createSession(session);
        Cookie sessionCookie = new Cookie("sid",
String.valueOf(sid));
        header.addCookie(sessionCookie);
        Cookie rememberMeCookie = new Cookie("rememberme", "on");
        header.addCookie(rememberMeCookie);
        header.addLocation("http://localhost:180/cgi-bin/home.cgi");
    }
}

private static void goToMain() {
    header.addLocation("http://localhost:180/cgi-bin/home.cgi");
}
```

- Invoke the **main** method

## SignIn.java

```
public static void main(String[] args) {
    readParameters();
    header.printHeader();
    readHtml();
}
```

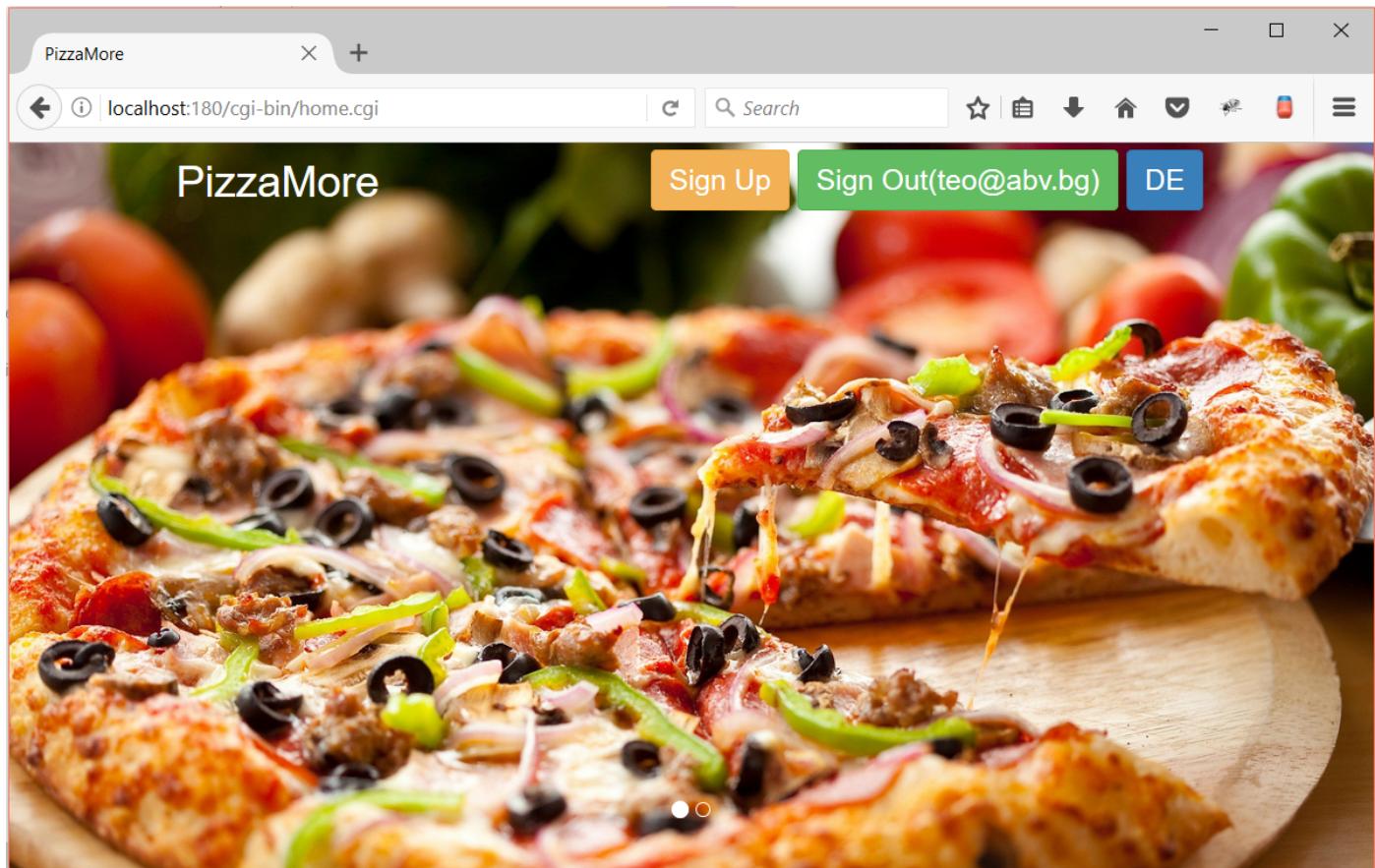
- Create a CGI file

## signin.cgi

```
#!/C:/cygwin64/bin/bash.exe
```

```
java -Dcgi.query_string=$QUERY_STRING -Dcgi.request_method=$REQUEST_METHOD -jar  
SignIn.jar
```

## 5. Read Session from Home Page



We need to update our home page and make it read session cookies

- Update **Home.java**

### Home.java

```
private static SessionRepository sessionRepository;  
...  
static {  
    ...  
    sessionRepository = new SessionRepository();  
}
```

- Read the **session cookie** and add change the **Sign In** button to **Sign Out**

## Home.java

```
private static void signOut(long id) {
    sessionRepository.delete(id);
}

private static void readHtml() {
    Cookie sessionCookie = cookies.get("sid");
    String username = null;
    if (sessionCookie != null) {
        long sid = Long.parseLong(sessionCookie.getValue());
        Session session = sessionRepository.findById(sid);
        //Sign Out
        if (parameters.containsKey("signout")) {
            signOut(sid);
            session = null;
        }

        if (session != null) {
            Set<SessionData> sessionData = session.getSessionData();
            for (SessionData data : sessionData) {
                if (data.getKey().equals("username")) {
                    username = data.getValue();
                }
            }
        }
    }

    Cookie languageCookie;
    if (!WebUtils.isPost()) {
        if (cookies.containsKey("lang")) {
            languageCookie = cookies.get("lang");
            if (languageCookie.getValue().equals("DE")) {
                readHtmlDe(username);
            } else {
                readHtmlEn(username);
            }
        } else {
            if (parameters.containsKey("language")) {
                String language = parameters.get("language");
                if (language.equals("DE")) {
                    readHtmlDe(username);
                } else {
                    readHtmlEn(username);
                }
            }
        }
    }
}

private static void readHtmlEn(String username) {
    String name = "signin";
    String value = "Sign In";
    if (username != null) {
        name = "signout";
        value = "Sign Out(" + username + ")";
    }
...
}
```