Web Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit

Building Web Apps

Web Development

Building a Web Application

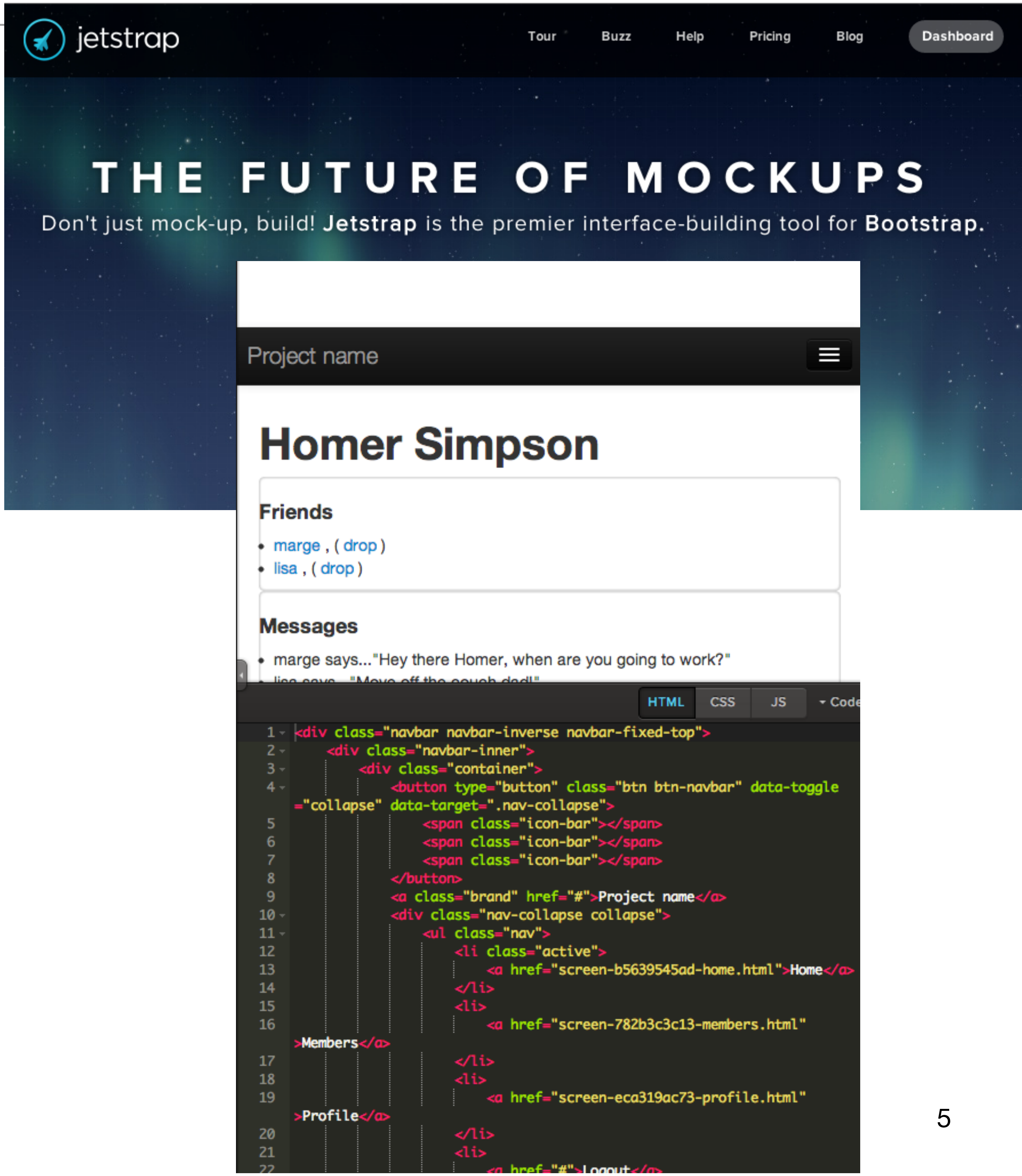
- First - build a web site:
  - Step 1: Determine Theme + Content
  - Step 2: Devise Navigation Structure
  - Step 3: Create Page Structure
  - Step 4: Apply a Style
  - Step 5: Build, Test & Deploy
- This site can be considered to be a ‘**Mock Up**’
  - Gives customer a feel for how the application will look
  - Allows customer to make changes before ‘development’ stage commences

With Mockup in place...Building a **Web App:**

- Step 1: Build Skeleton Web Application
- Step 2: Build Initial Templates
- Step 3: Analysis & Design -> Produce ‘Stories’
  - Step 4: Select a single ‘Story’
  - Step 5: Implement the ‘Story’
  - Step 6: Verify the implementation
- Step 7: Test & Deploy

Step 1: Build Skeleton Web Application

- To present a ‘static’ version of the site
- May look an feel identical to the pure html/css version built as a mock up
- However, the pages themselves will be composed of “Templates”
- Templates contain:
    - Conventional HTML
    - Special notations to enable ‘dynamic’ aspects of a page to be specified
- The CSS may remain unmodified from the ‘mock-up’ version.



THE FUTURE OF MOCKUPS

Don't just mock-up, build! **Jetstrap** is the premier interface-building tool for **Bootstrap**.

Homer Simpson

Friends

- marge , (drop)
- lisa , (drop)

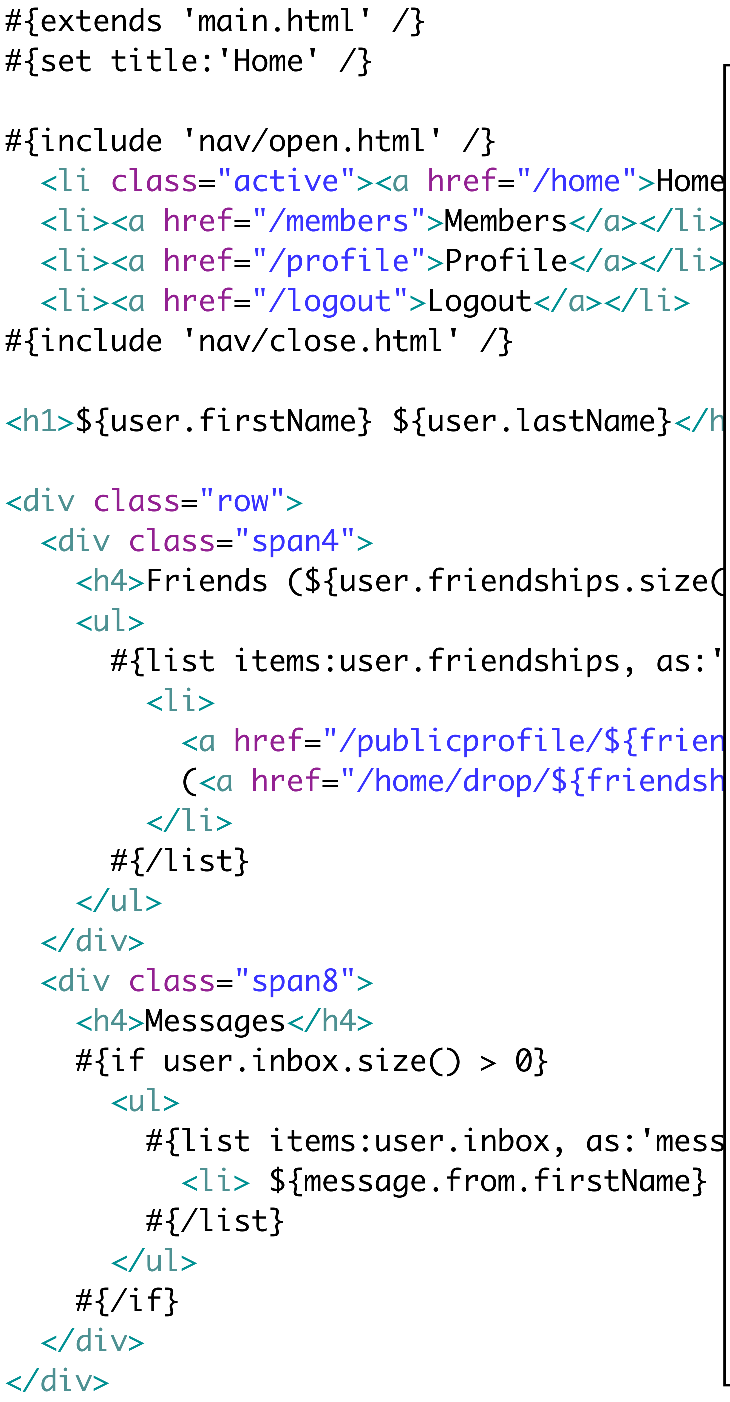
Messages

- marge says..."Hey there Homer, when are you going to work?"
- lisa says... "Move off the couch dad!"

```
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="navbar-inner">
        <div class="container">
            <button type="button" class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="brand" href="#">Project name</a>
            <div class="nav-collapse collapse">
                <ul class="nav">
                    <li class="active">
                        <a href="screen-b5639545ad-home.html">Home</a>
                    </li>
                    <li>
                        <a href="screen-782b3c3c13-members.html">Members</a>
                    </li>
                    <li>
                        <a href="screen-eca319ac73-profile.html">Profile</a>
                    </li>
                    <li>
                        <a href="#">Logout</a>
```

Step 2: Build Initial Templates

- Decompose the mock-up into ‘fragments’
- These fragments will represent parts of a page
    - Navigation Structure
    - + Various “views” or view fragments
- These fragments may be called ‘Templates’

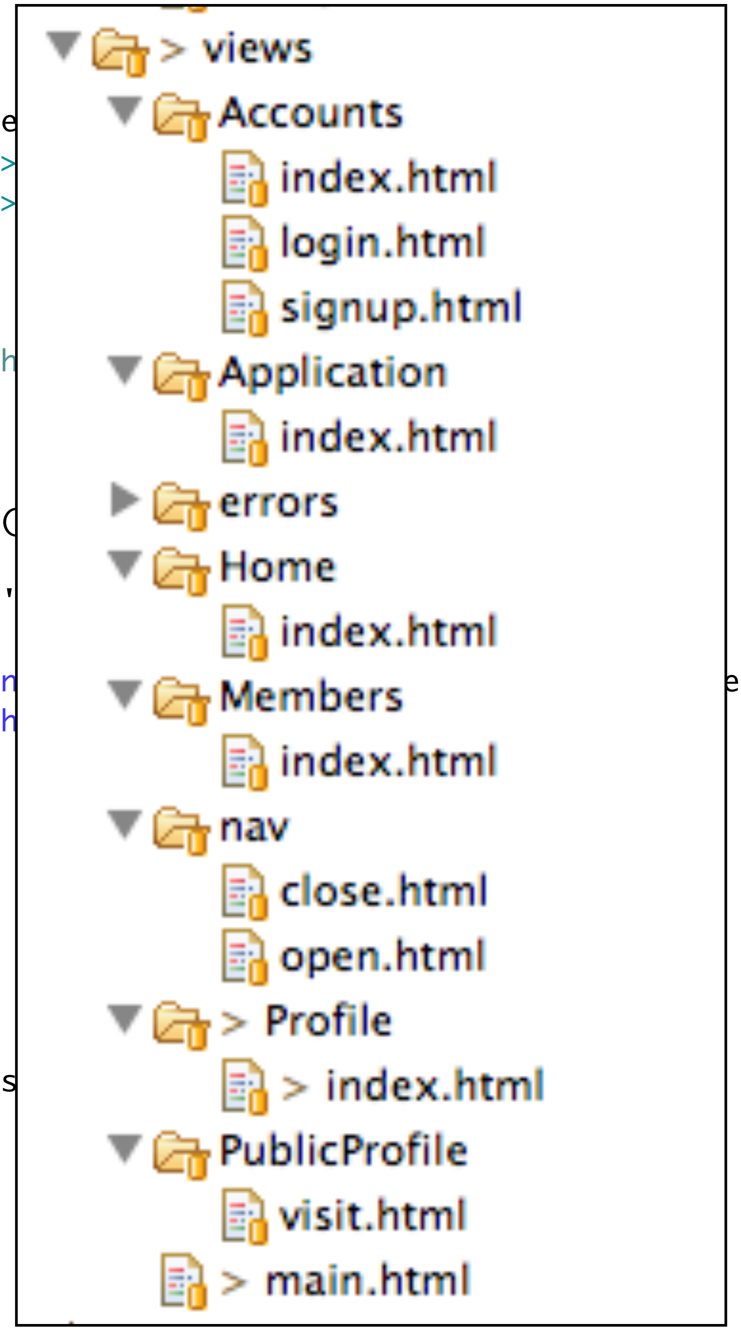


```
#{extends 'main.html' /}
#{set title:'Home' /}

#{include 'nav/open.html' /}
  <li class="active"><a href="/home">Home
  <li><a href="/members">Members</a></li>
  <li><a href="/profile">Profile</a></li>
  <li><a href="/logout">Logout</a></li>
#{include 'nav/close.html' /}

<h1>${user.firstName} ${user.lastName}</h

<div class="row">
  <div class="span4">
    <h4>Friends (${user.friendships.size(
    <ul>
      #{list items:user.friendships, as:'
        <li>
          <a href="/publicprofile/${frien
          (<a href="/home/drop/${friendsh
        </li>
      #{/list}
    </ul>
  </div>
  <div class="span8">
    <h4>Messages</h4>
    #{if user.inbox.size() > 0}
      <ul>
        #{list items:user.inbox, as:'mess
          <li> ${message.from.firstName}
        #{/list}
      </ul>
    #{/if}
  </div>
</div>
```



```
▼ > views
  ▼ Accounts
      index.html
      login.html
      signup.html
  ▼ Application
      index.html
  ▶ errors
  ▼ Home
      index.html
  ▼ Members
      index.html
  ▼ nav
      close.html
      open.html
  ▼ > Profile
      > index.html
  ▼ PublicProfile
      visit.html
    > main.html
```

etUse

Web App: Step 3: Analysis & Design

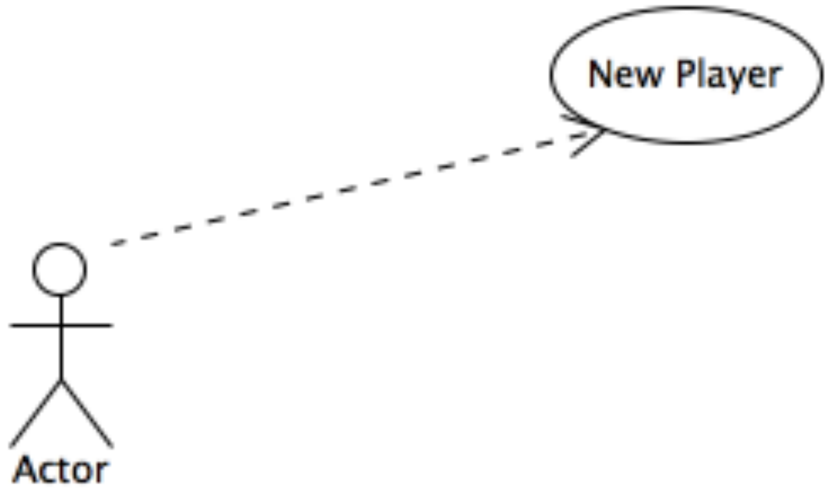
- Analyze the **User Requirements**, extracting a series of short **User Stories**

- Each story is a short statement describing some feature

- The story should be concise, observable to the end user, and its completion should be verifiable

- When all the stories are articulated, a **modeling** stage may commence

- Modeling may involve elaborating an initial design graphically

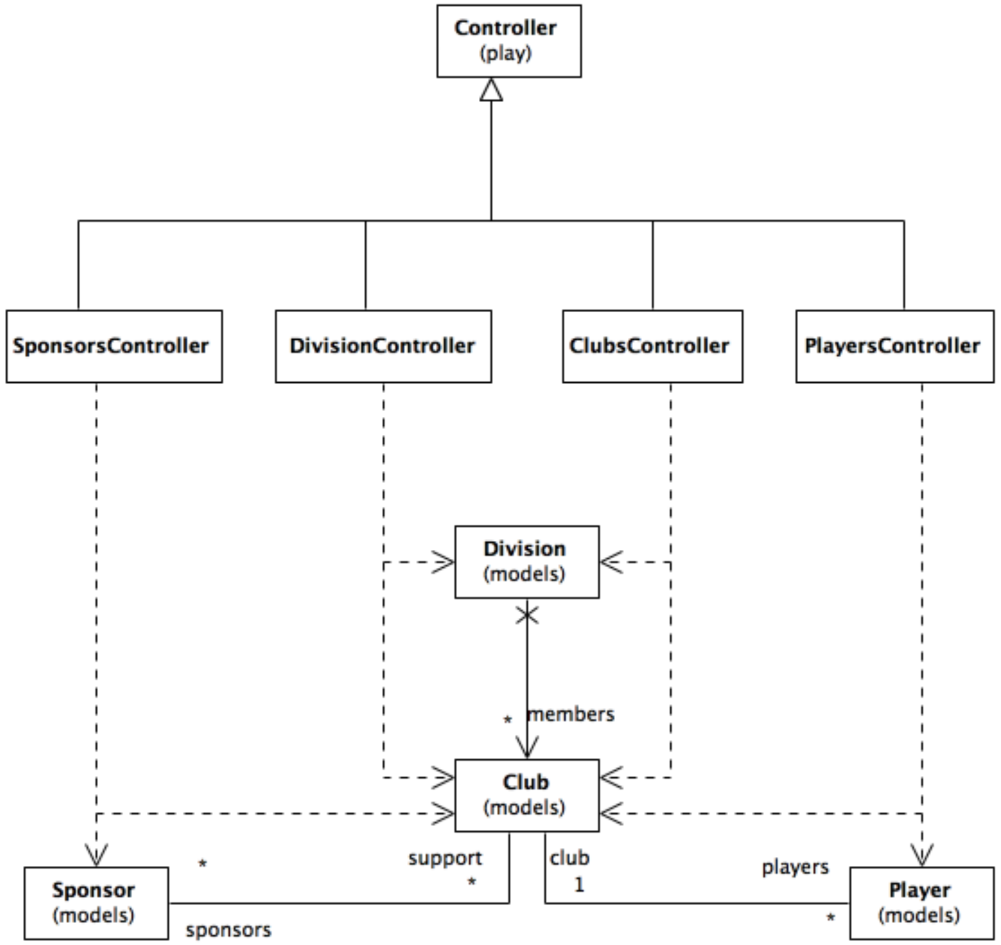


Web App: Step 4: Select a Single User Story

- The set of stories may be categorised into different groups.
- These groups may be further structured/ordered/sorted, so that they can be tackled sequentially.
- Select a story that can be realistically implemented in reasonable time.
- Review the story in the mock-up to understand it better

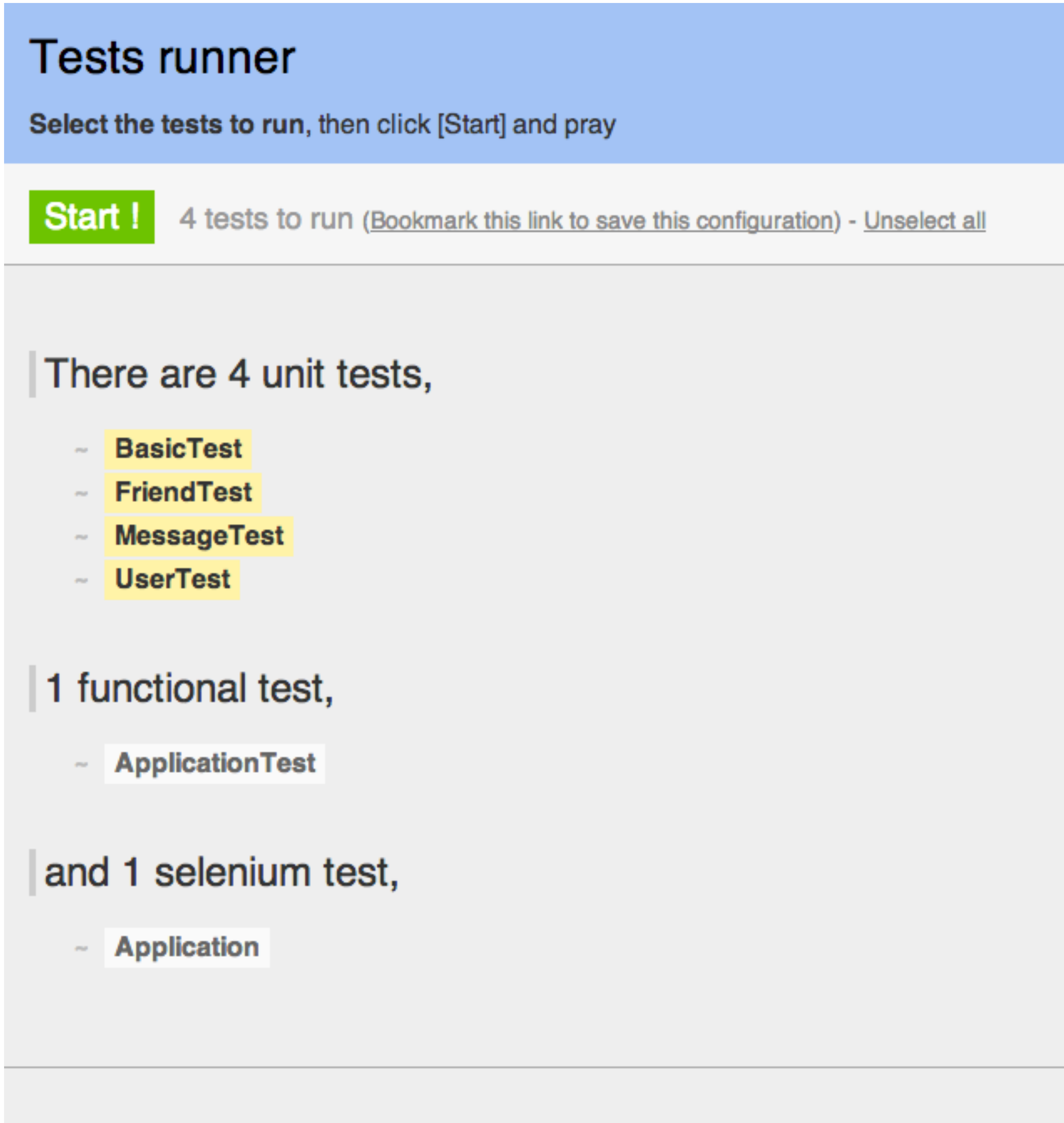
Web App: Step 5: Implement the Story

- Code sufficient capabilities in the HTML/CSS/Java sources to implement the story
- Implementing the store may involve:
  - Modifying the templates to facilitate user input
  - Writing Java code to realise new behaviour
  - Extending the Database to cater for new fields
  - Modifying the template to display new information



Web App: Step 6: Verify the Implementation

- Review the user story again
- Walk through the running application to verify that the story has concludes successfully
- Test 'Edge Cases' i.e.
  - incorrect or invalid data
  - blank data
  - unusual character combinations
- If any major anomalies arise, go back and fix them before moving on



Tests runner

Select the tests to run, then click [Start] and pray

4 tests to run (Bookmark this link to save this configuration) - Unselect all

There are 4 unit tests,

- BasicTest
- FriendTest
- MessageTest
- UserTest

1 functional test,

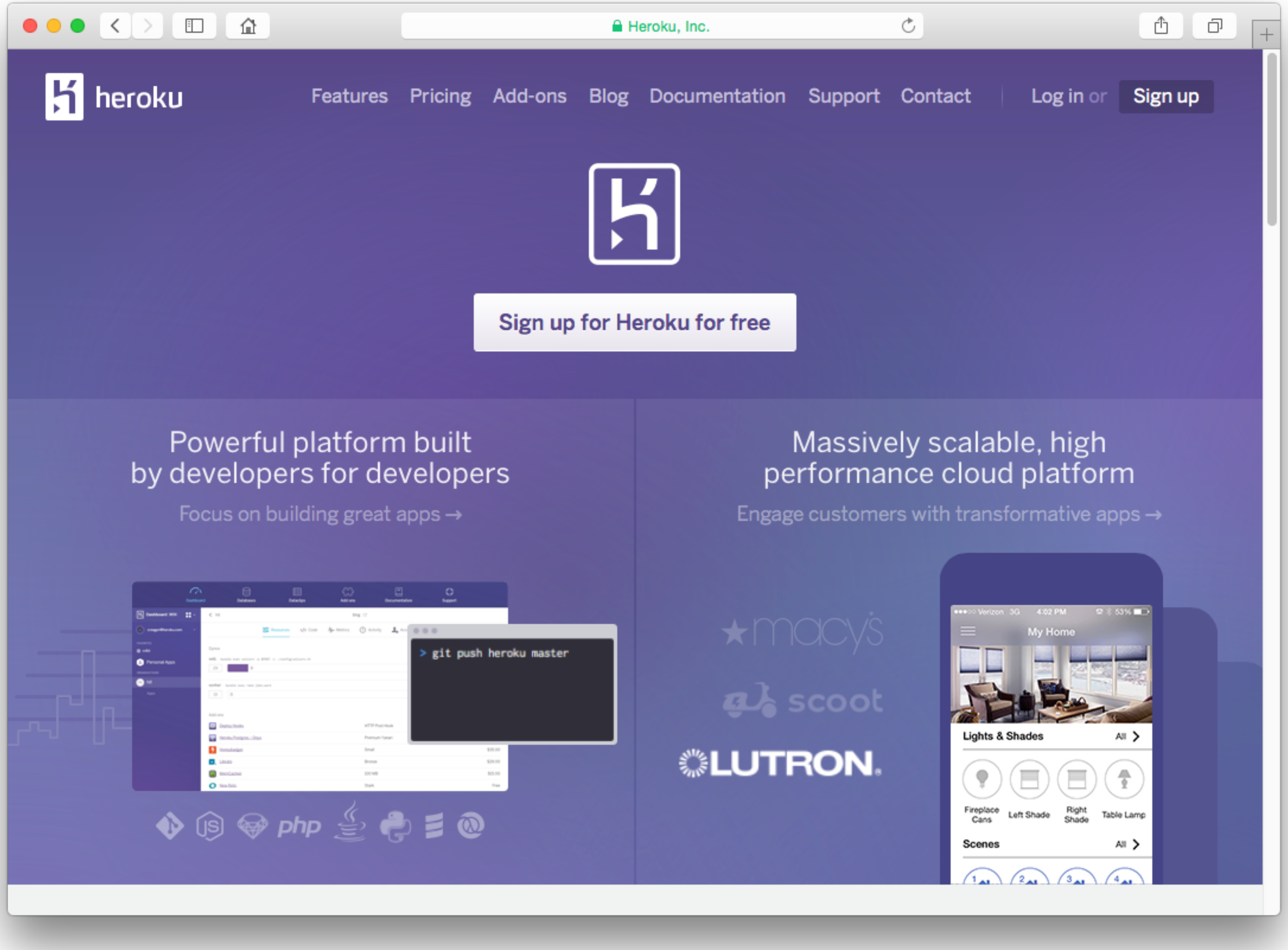
- ApplicationTest

and 1 selenium test,

- Application

Web App: Step 7: Deploy

- Publish the app to the web - or ‘the cloud’
- Verify that all of the user stories still work
- Handover to customer
- Maintenance!



Powerful platform built by developers for developers

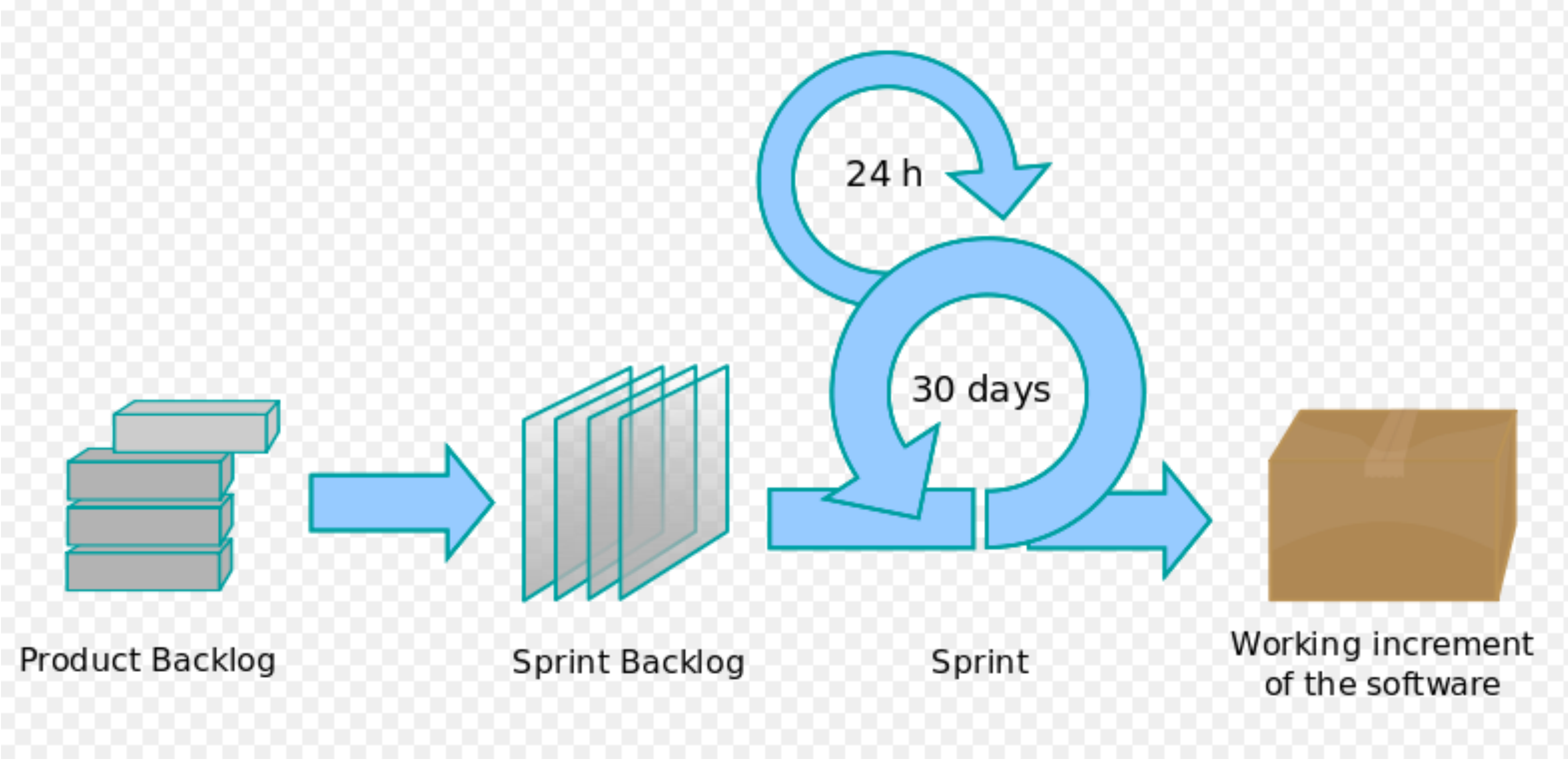
Focus on building great apps →

```
> git push heroku master
```

Massively scalable, high performance cloud platform

Engage customers with transformative apps →

Development Methodology - SCRUM



http://en.wikipedia.org/wiki/Scrum_(development)

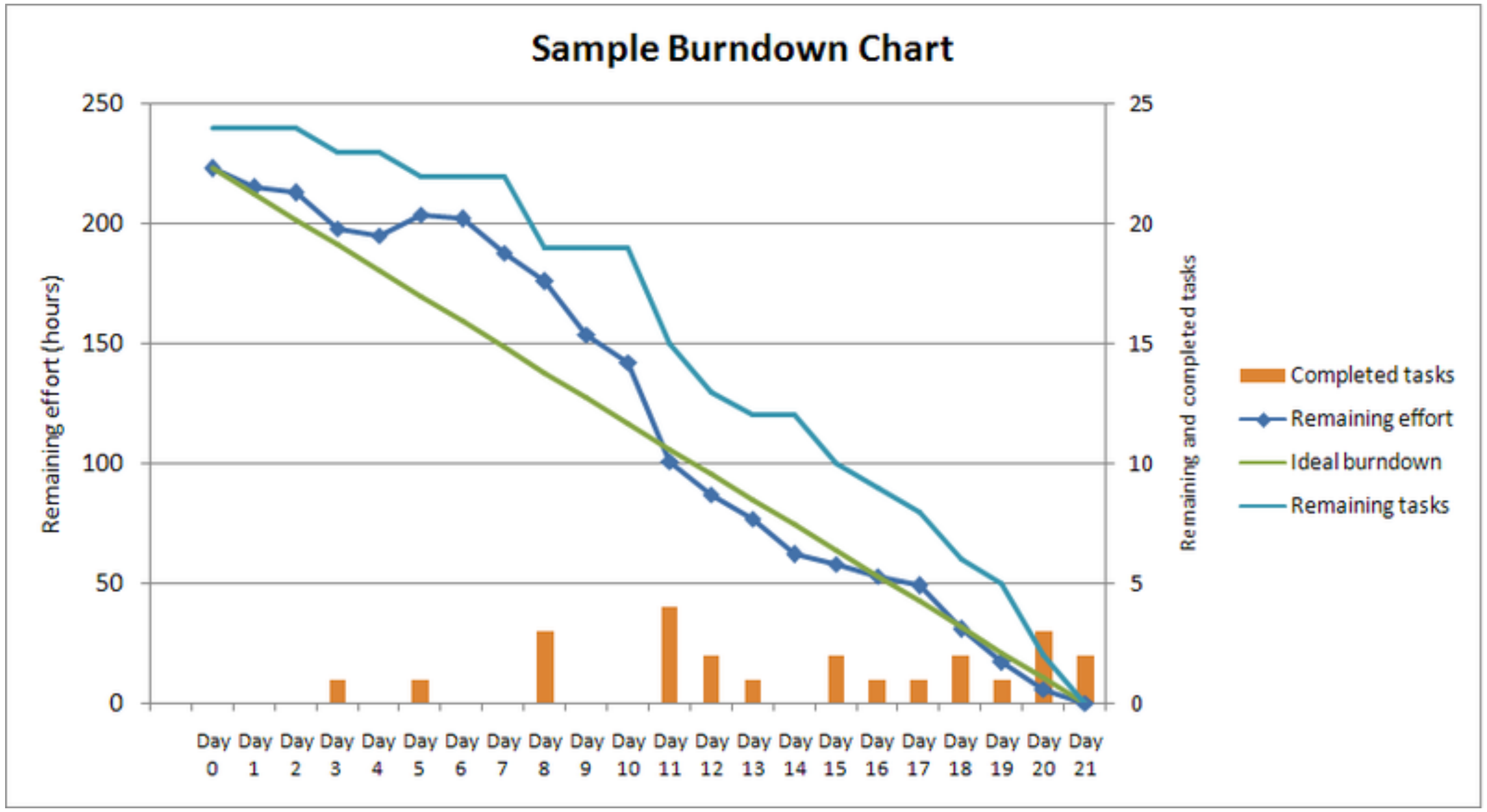
- Step 1: Build Skeleton Web Application
- Step 2: Build Initial Templates
- Step 3: Systems Analysis

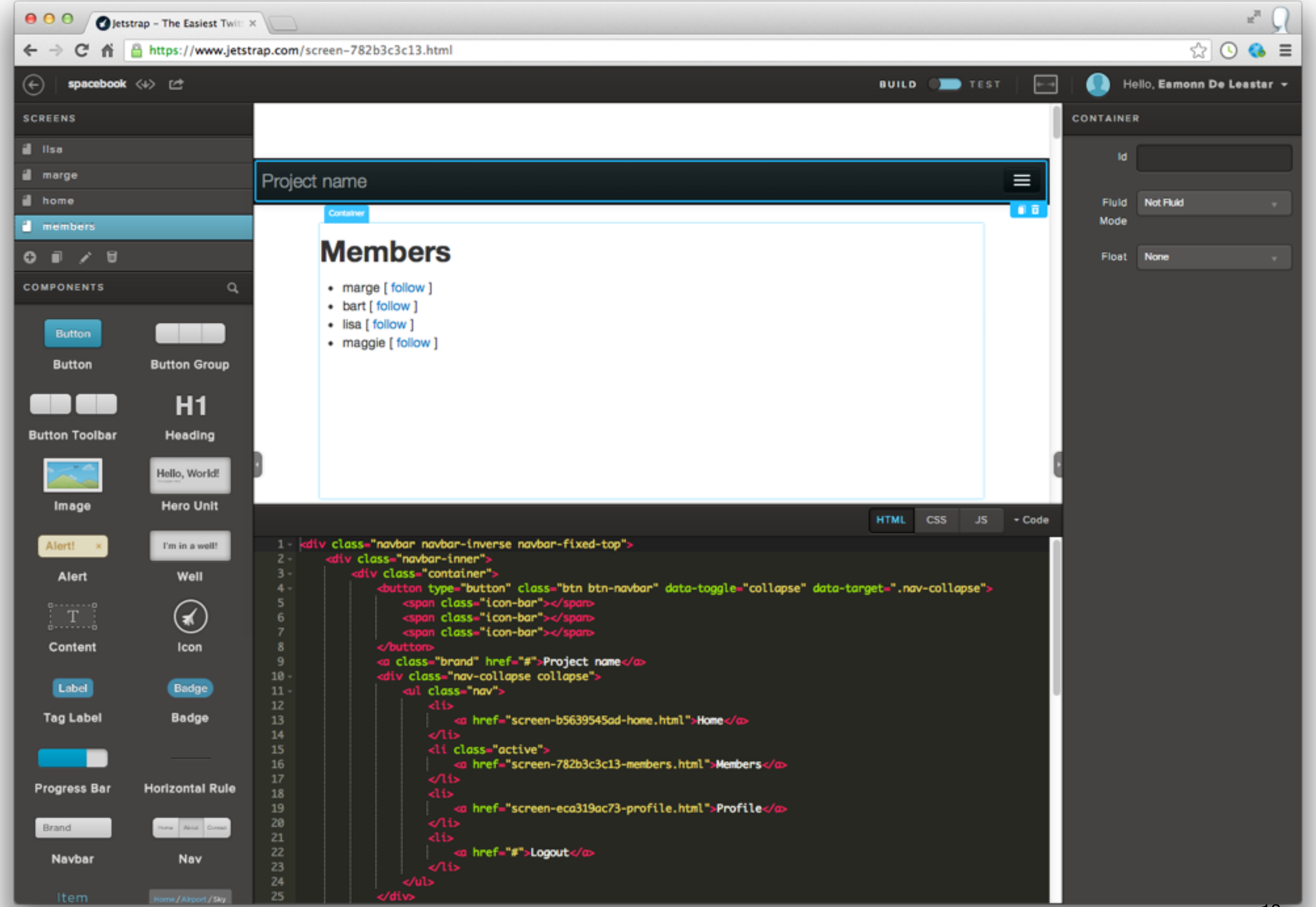
Step 4: Select a single User Story

Step 5: Implement the Story

Step 6: Verify the implementation

- Step 7: Deploy





Id

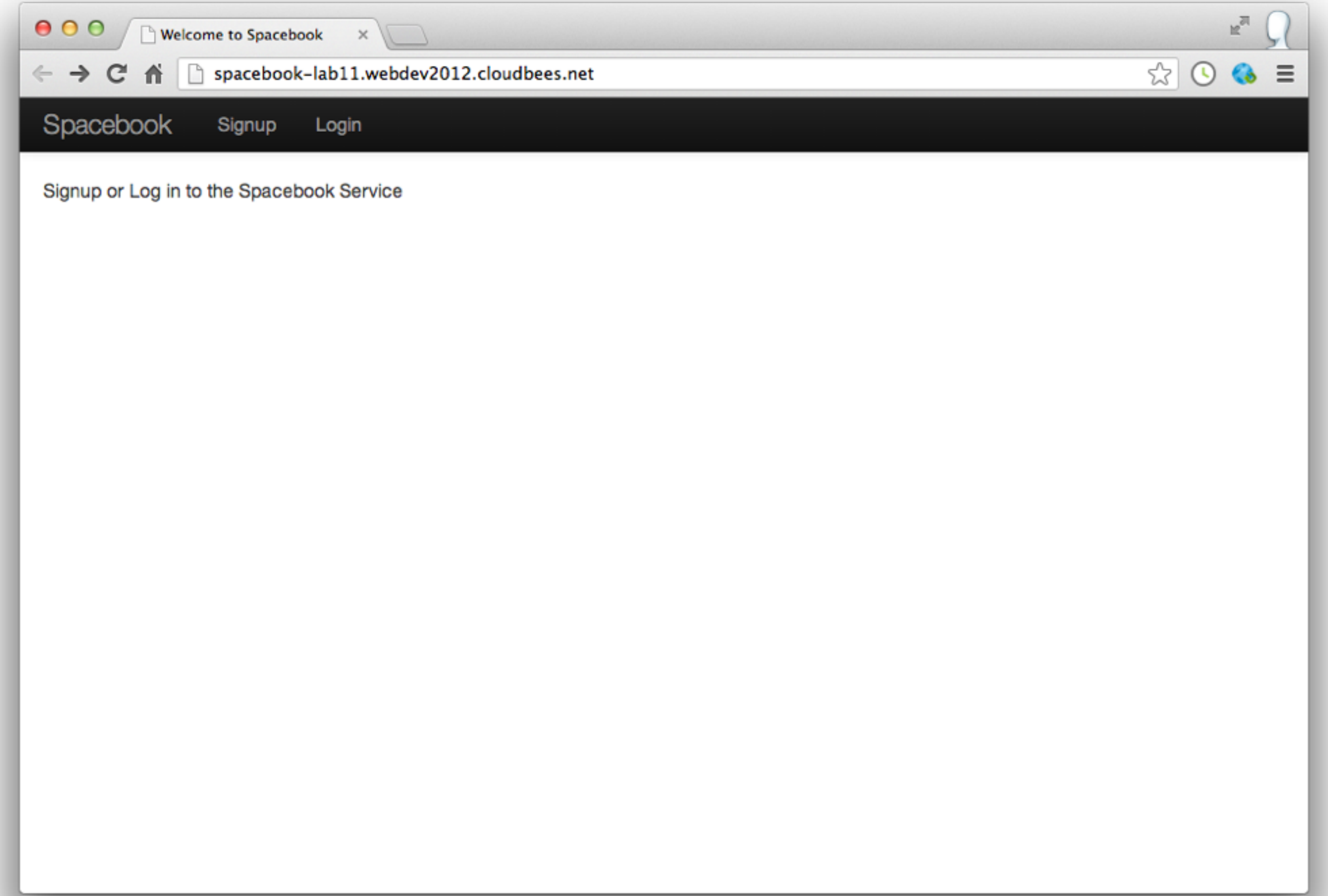
Fluid Mode: Not Fluid

Float: None

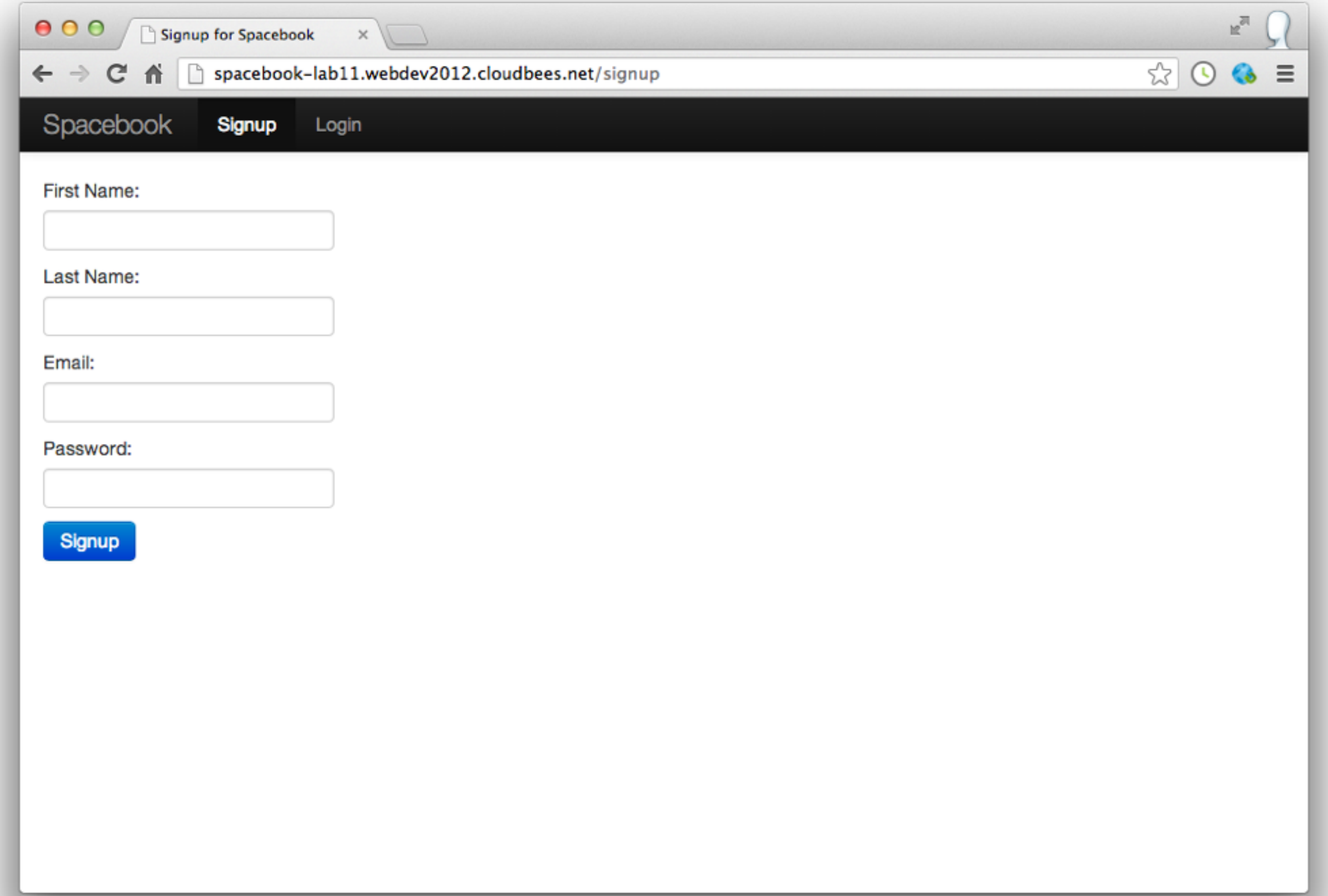
Members

- marge [follow]
- bart [follow]
- lisa [follow]
- maggie [follow]

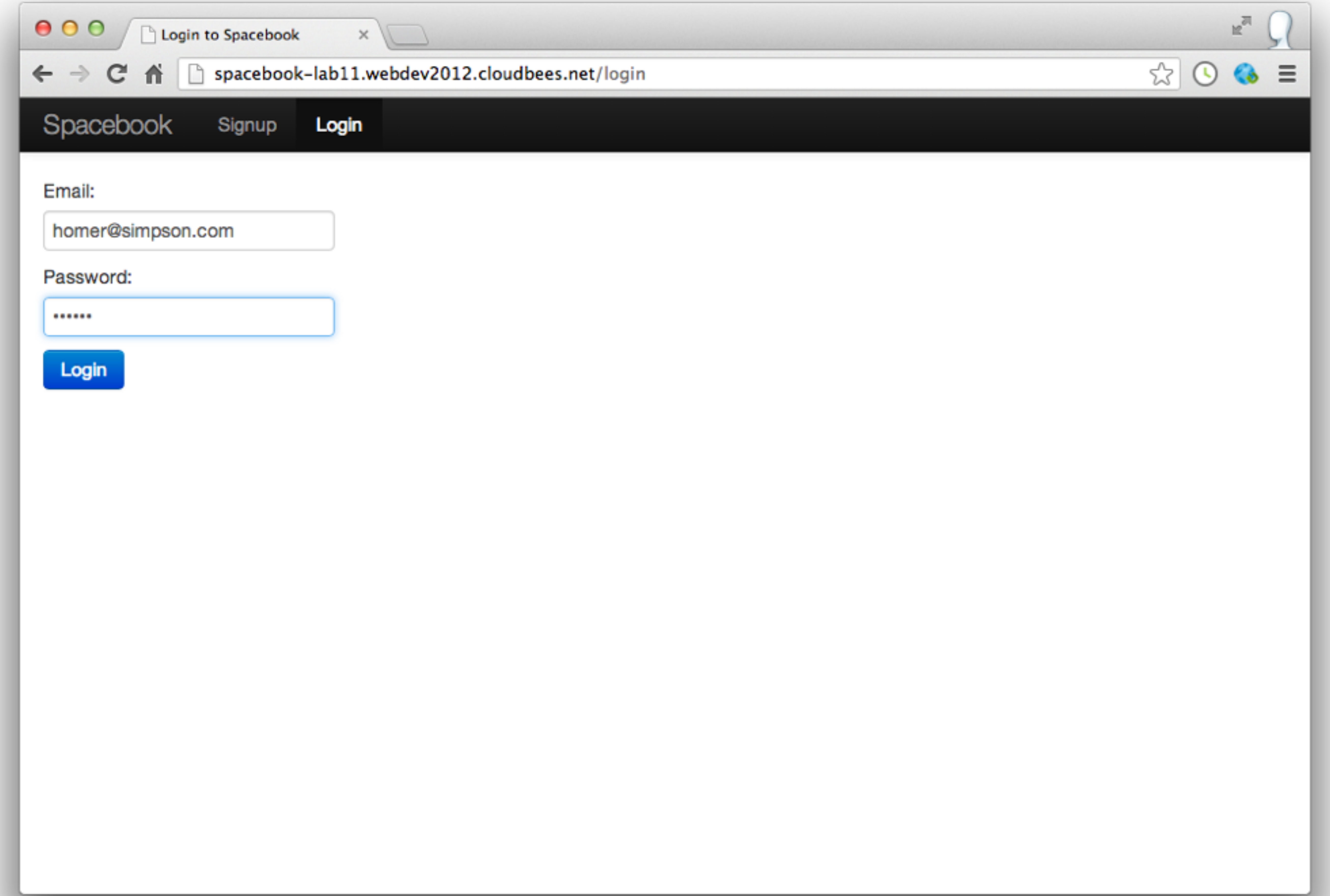
```html
<div class="navbar navbar-inverse navbar-fixed-top">
    <div class="navbar-inner">
        <div class="container">
            <button type="button" class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="brand" href="#">Project name</a>
            <div class="nav-collapse collapse">
                <ul class="nav">
                    <li>
                        <a href="screen-b5639545ad-home.html">Home</a>
                    </li>
                    <li class="active">
                        <a href="screen-782b3c3c13-members.html">Members</a>
                    </li>
                    <li>
                        <a href="screen-eca319ac73-profile.html">Profile</a>
                    </li>
                    <li>
                        <a href="#">Logout</a>
                    </li>
                </ul>
            </div>
```



Signup or Log in to the Spacebook Service



Signup for Spacebook
spacebook-lab11.webdev2012.cloudbees.net/signup
Spacebook
Signup
Login
First Name:
Last Name:
Email:
Password:
Signup

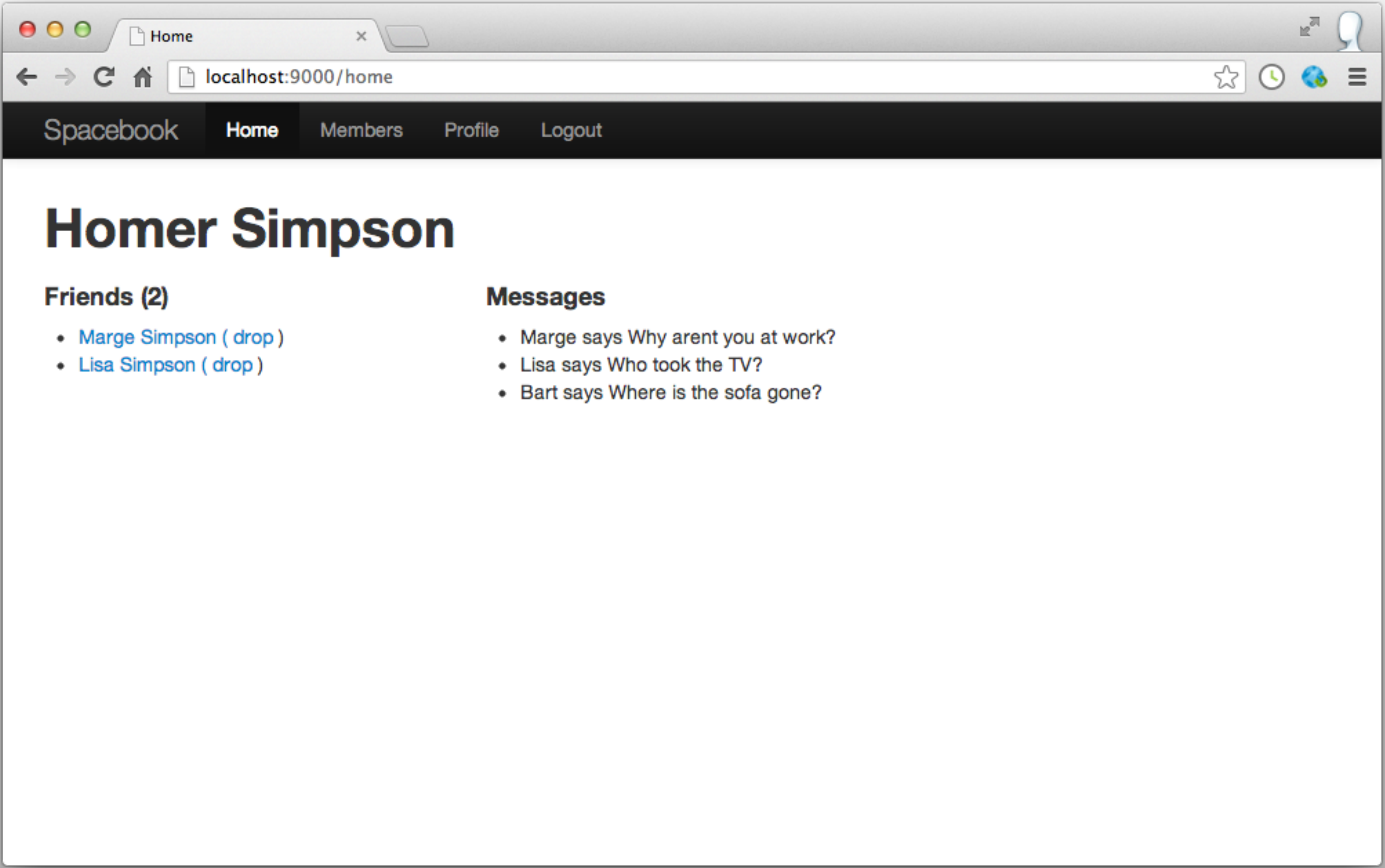


Email:

homer@simpson.com

Password:

••••••



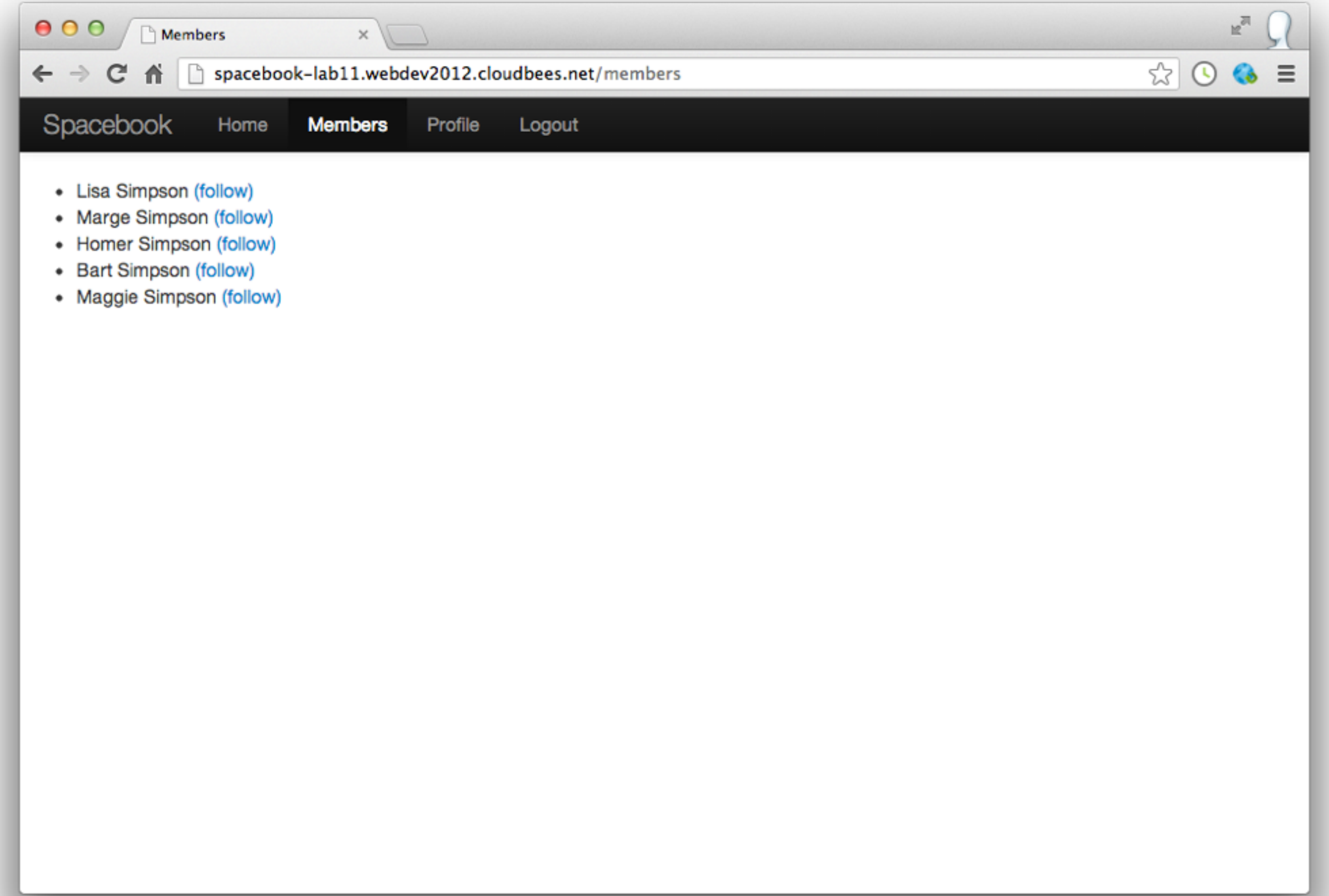
Homer Simpson

Friends (2)

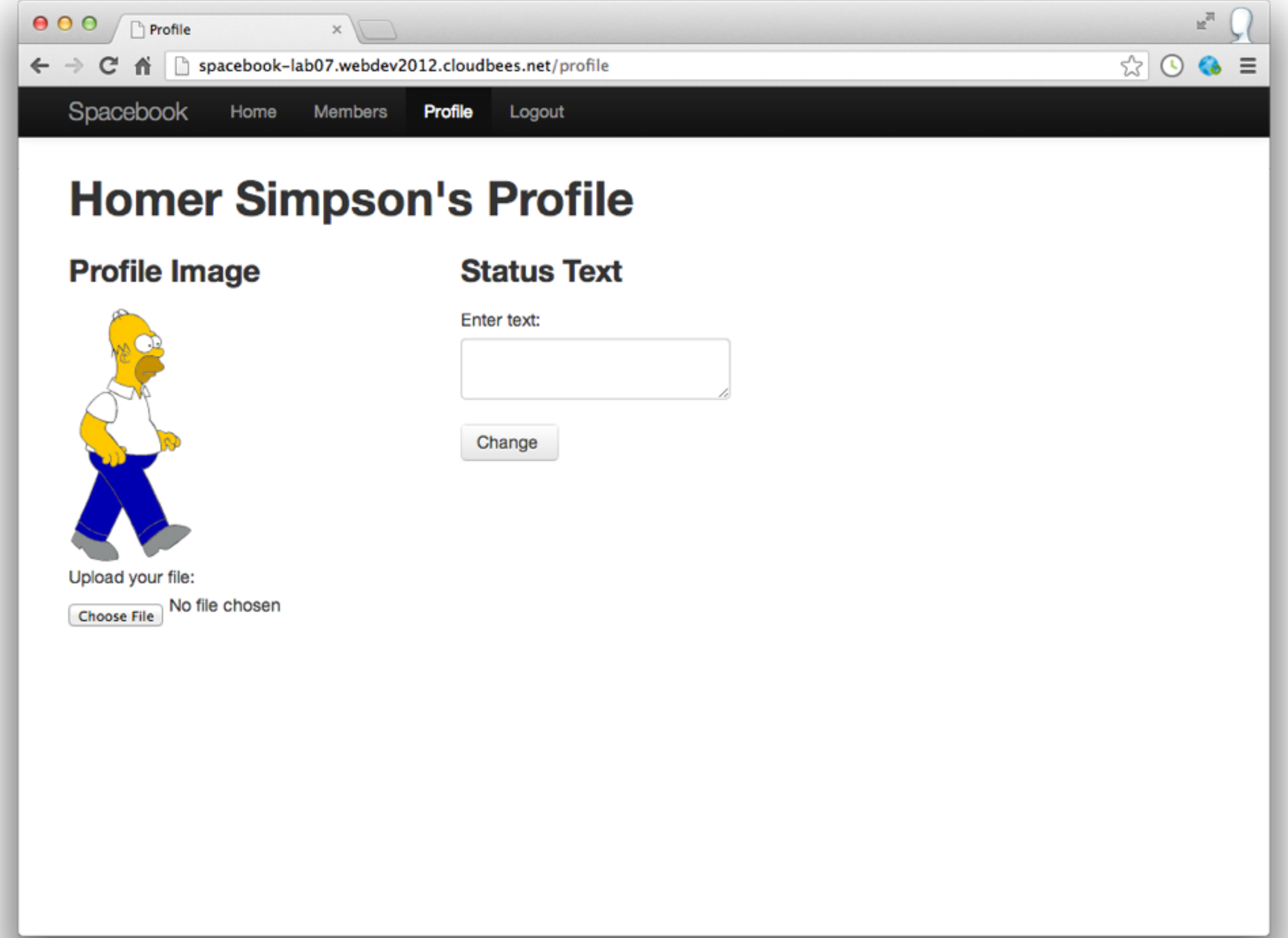
- Marge Simpson (drop)
- Lisa Simpson (drop)

Messages

- Marge says Why arent you at work?
- Lisa says Who took the TV?
- Bart says Where is the sofa gone?



- Lisa Simpson (follow)
- Marge Simpson (follow)
- Homer Simpson (follow)
- Bart Simpson (follow)
- Maggie Simpson (follow)



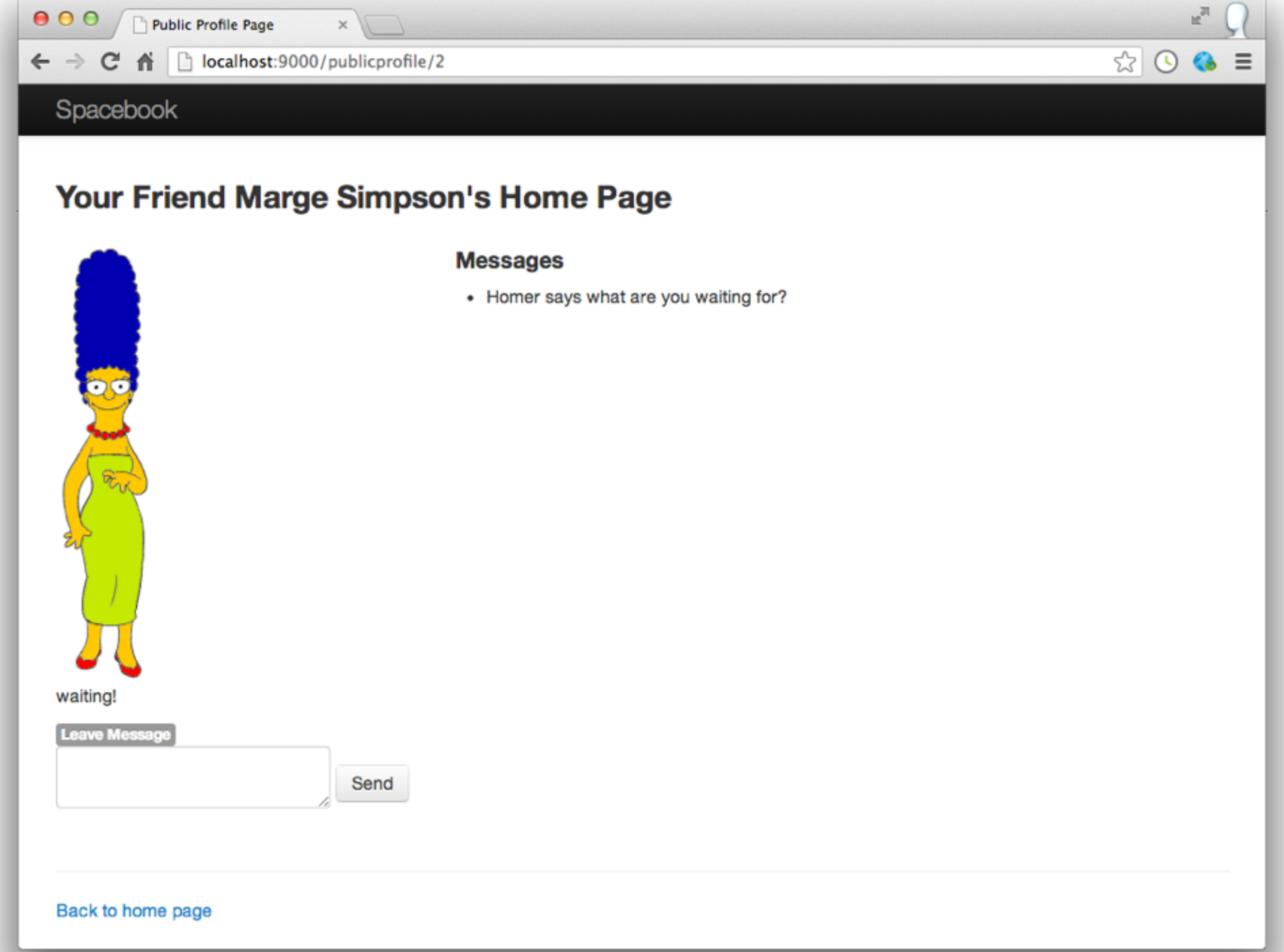
Homer Simpson's Profile

Profile Image

Status Text

Enter text:

Upload your file:



Your Friend Marge Simpson's Home Page

Messages

- Homer says what are you waiting for?

waiting!

Stories

Candidate Stories (1 of 2)

1. Present a tabbed navigation display with “Home”, “Members”, “Profile” and “Logout” tabs.
2. Before presenting the tabbed navigation, present a “Start” screen which will allow users to either sign up or log in
3. If users choose to sign up, let them register a new username and password on a “Signup” screen
4. If they choose to log in, let them validate username/password on a “Login” screen and, if they match an entry, let them see their “Home” page
5. A users “Home” and “Profile” pages should display their name prominently
6. A users “Home”page should display any messages left by other members for that user
7. From the “Profile” page, be able to upload new profile image which will become the default image for their public profile

Candidate Stories (2 of 2)

8. Be able to add a member as a friend. If the member is not a friend, then they will be listed as someone the user is ‘Friends’ with

9. Be able to view a friends profile by clicking on their name on the home page. It takes us to a “PublicProfile” view - which consists of the friends name, profile image + any messages already left for them by other users

10. From a users ‘FriendProfile” page, be able to leave a message for that member

11. From the “Home” page, be able to drop a friend

12. On the “Members” page, the list of members should not include the currently logged in member

13. Be able to log out

New Features?

1.Let user delete his/her account

2.Allow a friend to leave a private message - only visible to the friend (not on public profile)

3.As well as profile text, introduce concept of ‘status’. Be able to see status from list of friends

4.Be able to see whether friends are currently logged in or not

5.Chat room feature, where multiple friends can join a single room

7.Friend to Friend Chat

8.Photo library (as opposed to single profile image)

9.Public and private photo albums

10.Delete messages

11.Archive messages

12.Search messages

13. Blog

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit