Funtown Project Review (加封面于此，并加到另外一个文件)

4.1 Project Journal

Here are the journal composed by 12 weekly reports:

**7160 Weekly Report Chang Shu 18613 (May 20 - May 26)**

Main dicussion about my previous work on the itemDetail page.

Following work assigned.

Aaron's suggestion and mission:

Modify my style on the font, size, line break, margin of the main body;

Modify the style of headers;

Change the colour of R icon;

Always give a large &lt;div&gt; a clear class name or names;

Try to use sass syntax to write scss files. put subclass styles into the curve brackets of main classes.

Use class="ratebox ratebox-R" to apply the general style and specified styles respectively.

remove the "active" class name. Every class name without specific purpose should be removed.

Rename all the class names following the rules above.

Aaron's instruction:

The name of classes should not follow the CamelCase rules. We should use lowercase words linked by dash

Requirement:

So far, we give up on the compatibility of mobile devices. This website should display correctly on iPads, laptops and desktops only.

**7160 Weekly Report Chang Shu(May 27 - Jun 2)**

**Aaron's suggestion & mission**

◆Always remember to use 4 spaces for indentation, instead of tabs! Reason: the spaces have much better compatibility in different compilers

◆In my html files, should break lines between different elements/tagnames.

◆Need to correct the logic in my js code:

Chech our code in the following file

D:\funtown\funtown-web\webapp\app\modules\selling\controllers\product-add.controller.js:

The product id is get by “$scope.pid = $state.params.pid;”

I shall make the url of my itemdetail page as “itemDetail?pid=1”

About our project: Efficiency and teamwork

◆The progress of our project is slow.

◆Solution: We need to bring in standardized and industrialized developing procedures

◆Solution: We need to designate someone to follow each team member’s progress, set deadlines and ensure the whole project’s running

**7160 Weekly Report Chang Shu 18613 Meeting 11 (Jun 3 - Jun 9)**

Wrote three functions for product.service:

1. Return a property value from an object item with another specified property value:

this.findAttributeType = function (attriType, arr) {

for (var i = 0; i < arr.length; i++) {

if (arr[i].attributeType == attriType){

return arr[i].attributeValue;

}

}

}

2. Sorting an array by a specified property value:

this.reorderArreyByAttributeId = function (arr) {

arr.sort(function(a, b){return a.attributeId - b.attributeId});

}

3. Soft copy an array:

this.cloneArray = function (arr) {

var cloneOfA = JSON.parse(JSON.stringify(arr));

return cloneOfA;

}

About our project: Efficiency and teamwork

A deeper understanding of Git:

git command for config checking: git config --list

display the changes that have been made: git diff

[important]add all files to staging area: git add -A

[important]make a commit: git commit -m "initial commit"

[important]push my master branch to the origin project: git push origin master

check my commit log: git log

display all the branch info of this repository: git branch

display all the branch info of this repository, including the remote ones: git branch -a

[important]working on branches - create a branch: git branch charles-branch

[important]working on branches - move to a branch: git checkout charles-branch

[important]working on branches - after adding and commiting, push my branch to remote: git push -u origin charles-branch

bash command for listing every file: ls -la

**7160 Weekly Report Chang Shu 18613 (Jun 10 - Jun 16) report**

About our project: Kedar and Jessica are joining our team and making the progress faster.

A deeper understanding and use of Git:

About git add: “git add -A” is equivalent to “git add .”. These two commands add all modified and new (untracked) files in the current directory and all subdirectories to the staging area (a.k.a. the index), thus preparing them to be included in the next git commit

Set an alias for complicated git commands: $ git config --global alias.st status

The difference between ‘git pull’ and ’git rebase’: when you pull your code from dev branch to master branch, it will only generate one commit on the master branch declaring all the modifications on the dev branch; while when you rebase your code from dev branch to master branch, it will bring all those commits on the dev branch to this master branch. In a nutshell, use ‘git rebase’.

Three ways to make a change on a previous commit:

1. Git revert: Reverting undoes a commit by creating a new commit. This is a safe way to undo changes, as it has no chance of re-writing the commit history.

2. Git reset: move the head to the location where when the last change of this commit is done, also the next modifications in the following commits will EXIST as unstaged changes.

3. Git checkout: This git checkout <commit> command can let you enter a 'detached HEAD' state to the commit position, where when the last change of this commit is done. You can look around, make experimental changes and commit them, **and you can discard any commits you make in this state without impacting any branches by performing another checkout**.

Three distinct uses of ‘git checkout’:

1. git checkout <branch>: To prepare for working on <branch>, switch to it by updating the index and the files in the working tree, and by pointing HEAD at the branch.

2. git checkout <filename>: For those changed file that is TRACKED, i.e. changes that is not staged for commit yet, using this command will ERASE all the changes in working directory.

For those files that are already staged, “git checkout <filename>” is useless. We can only use "git reset HEAD <file>" to unstage this file. And this command "git reset HEAD <file>" will not change this file, only unstage it and put it back to “Tracked files: changes not staged for commit” or “untracked files”

3. git checkout <commit>: Enter a 'detached HEAD' state to the commit position, where when the last change of this commit is done. You can look around, make experimental changes and commit them, **and you can discard any commits you make in this state without impacting any branches by performing another checkout**.

About “console.log”

[important] console.log is always a little "late" and you can't count on it when it comes to objects. Only primitives (strings etc.) will work directly. Of the former there is only one instance in memory, so when the console is fetching the data it may have changed already.

Solution: always remember to use $log.debug instead.

Some concepts about Git:

Head: the default position you are working with

Origin: representation of the coding

Master: the main branch of the product

Aaron’s advice:

Always use standard and formal sentences in the commit, starting with a verb. Never use a sentence like “hope it will work”.

Assignment:

Study through the “git fetch”

**7160 Weekly Report Chang Shu 18613 Meeting 12 (Jun 17 - Jun 23) report**

There are two ways to create an app on Heroku:

1. Add config files based on the tutorial with this URL: https://www.youtube.com/watch?v=P86N9FqNqso&amp;index=2&amp;list=LLL\_3tdfMBxvgcKjMzUfyy7w; then navigate to the "add app" GUI page; finally link my website from github or gitlab to Heroku
2. Add config files based on the tutorial with this URL: https://www.youtube.com/watch?v=P86N9FqNqso&amp;index=2&amp;list=LLL\_3tdfMBxvgcKjMzUfyy7w; then use the Heroku command line to start the app: https://devcenter.heroku.com/start?c=https://devcenter.heroku.com/start&amp;utm\_campaign=Onboarding-Nurture-Email-1&amp;utm\_medium=email&amp;utm\_source=nurture&amp;utm\_content=devcenter&amp;utm\_term=start-all

There are two ways to edit/test Heroku apps:

1. Use the Heroku command line tools:

Input <code>heroku log in</code>

type in the credentials

Input <code>heroku local web</code>

Enter the address localhost:5000

and input this

Note: this is for convinent local edit and test only

1. Put the code and see the page on Internet:

Use the procedures above, in the Assignment on Jun 14th

Push the code after modification

Important concept about back-end programming:

Apache

Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all webservers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many different environments by using extensions and modules. Most WordPress hosting providers use Apache as their web server software. However, WordPress can run on other web server software as well.

Node.js

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

Express.js

Express.js is a Node.js framework. It's the most popular framework as of now. Some say express.js is to Node.js what Ruby on Rails or Sinatra is to Ruby. Express 3.x is a light-weight web application framework to help organize your web application into an MVC architecture on the server side. You can use a variety of choices for your templating language (like EJS, Jade, and Dust.js).

Maven

Maven is a build automation tool used primarily for Java projects.

Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies. Contrary to preceding tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins.

Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation.

npm

npm, short for Node Package Manager, is two things: first and foremost, it is an online repository for the publishing of open-source Node.js projects; second, it is a command-line utility for interacting with said repository that aids in package installation, version management, and dependency management.

Spring

The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform.

JPA

The Java Persistence API (JPA) is a Java specification for accessing, persisting, and managing data between Java objects / classes and a relational database. JPA was defined as part of the EJB 3.0 specification as a replacement for the EJB 2 CMP Entity Beans specification. JPA is now considered the standard industry approach for Object to Relational Mapping (ORM) in the Java Industry.

JPA itself is just a specification, not a product; it cannot perform persistence or anything else by itself. JPA is just a set of interfaces, and requires an implementation. There are open-source and commercial JPA implementations to choose from and any Java EE 5 application server should provide support for its use. JPA also requires a database to persist to.

REST

REST is an architecture style for designing networked applications. The idea is that, rather than using complex mechanisms such as CORBA, RPC or SOAP to connect between machines, simple HTTP is used to make calls between machines.

In many ways, the World Wide Web itself, based on HTTP, can be viewed as a REST-based architecture.

RESTful applications use HTTP requests to post data (create and/or update), read data (e.g., make queries), and delete data. Thus, REST uses HTTP for all four CRUD (Create/Read/Update/Delete) operations.

REST is a lightweight alternative to mechanisms like RPC (Remote Procedure Calls) and Web Services (SOAP, WSDL, et al.). Later, we will see how much more simple REST is.

Despite being simple, REST is fully-featured; there's basically nothing you can do in Web Services that can't be done with a RESTful architecture.

**7160 Weekly Report Chang Shu 18613 Meeting 14 (Jul 2 - Jul 8) report**

Important usage of cherry-pick,merge and rebase

• cherry-pick: copy the selected [commit] to the current branch. find more details from here.

Note: you need to manually delete the original commits if moving is your purpose.

• git merge:[important! This may be the one I need most] use git merge branchname to merge this "branchname" into current branch. Remember: use git add and git commit to solve conflicts.

find more details from here.

• git rebase: use git rebase branchname to move all the commits from "branchname" to top of the current branch.

Note 1: after this process, the original branch "branchname" still exists!

Note 2: We usually use git merge to merge feature branch to master branch; while use git rebase on a feature branch to linearize the relation between master branch and feature branch and then we use git merge to move everything back to master branch.

find more details from here.

The Golden Rule of git rebase

Never use git rebase from the master branch! Try to use git rebase on a feature branch to linearize the relation between master branch and feature branch and then we use git merge to move everything back to master branch.

find more details from here.

Note: those material above is in Chinese, if you want to view an English version (and it is the best version), click HERE.

About the collapse function of Bootstrap

Collapse is even more beautiful than the ng-show of AngularJS.

To use it, remember to add these files:

<link rel="stylesheet" type="text/css" href="assets/lib/bootstrap/bootstrap-4.0.0-alpha.6-dist/css/bootstrap.css">

<script src="assets/lib/jquery/3.2.1/jquery-3.2.1.min.js"></script>

<script src="assets/lib/bootstrap/bootstrap-4.0.0-alpha.6-dist/js/bootstrap.js"></script>

Then download the newest version of bootstrap and jquery, and put them into corresponding folders.

Useful tip:

Use white-space: pre-wrap; to perfectly display a block of code!

A perfect site to study css-stripes! Click Here

For some html elements, use display: inline; to prevent them from automatically changing lines.

After use git checkout [commit] to visit a historical commit, I can use git checkout master (or [branch name]) to return the HEAD to the tip of workflow

**7160 Weekly Report Chang Shu 18613 Content (Jul 8 - Jul 14)**

Concerning JIRA:

Official site of Atlassian:

https://www.atlassian.com/software/jira

Some beginner's tutorial of JIRA

http://www.guru99.com/jira-tutorial-a-complete-guide-for-beginners.html

https://confluence.atlassian.com/get-started-with-jira-software/get-started-with-jira-software-844502163.html

Some JIRA agile tutorial

<https://confluence.atlassian.com/agile/jira-agile-user-s-guide/jira-agile-tutorials>

About JIRA

The #1 software development tool used by agile teams

**JIRA includes scrum software development and kanban software development**

Kanban vs Scrum vs Agile

http://www.agileweboperations.com/scrum-vs-kanban

Some concepts

user story

A user story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

Story point

Story point is an arbitrary measure used by Scrum teams. This is used to measure the effort required to implement a story. In simple terms its a number that tells the team how hard the story is. Hard could be related to complexity, Unknowns and effort.

ReactJS study note:

[important]ReactJS environment setup (official tutorial): https://www.tutorialspoint.com/reactjs/reactjs\_environment\_setup.htm

NOTE:

Whenever you want to use something, you need to import it first. If you want to make component usable in other parts of the app, you need to export it after creation and import it in the file where you want to use it.

7160 Weekly Report Chang Shu 18613 Content (Jul 15 - Jul 21)

Npm:

npm is the package manager for JavaScript

a much friendlier tutorial:https://facebook.github.io/react/tutorial/tutorial.html#getting-started

PHP:

[important]php environment setup (official tutorial)

https://www.tutorialspoint.com/php/php\_environment.htm

[important]beginner's Python tutorial

<https://www.tutorialspoint.com/python/python_overview.htm>

Prepare for interviews:

Front-End Job Interview Questions

https://github.com/yangshun/tech-interview-handbook/blob/master/front-end/interview-questions.md

Here is my study about these questions:

HTML interview questions:

TODO: difference between cookies and sessions

CSS interview questions:

z-index only effects elements that have a position value which is NOT static.

A good complete explanation of stacking context: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\_Positioning/Understanding\_z\_index/The\_stacking\_context

Always remember that position: absolute and position: fixed will REMOVE the element from the flow of the page

JS interview questions:

TODO

I successfully upgrade the node.js on windows:

Detecting if you're vulnerable from the Heroku CLI, i.e. checking whether the current node.js version has vulnerability problems:

heroku run node -v -a APPNAME

Find out that my current version is old and has a high severity remote Denial of Service (DoS) Constant Hashtable Seeds vulnerability.

Add these code into my package.json to specify the version of Node.js to use on Heroku:

"engines": {

"node": "6.11.1"

},

**7160 Weekly Report Chang Shu 18613 Content (Jul 22 - Jul 28)**

Node.js study note

Important Concepts:

Node.js is an open source server framework.

Node.js allows you to run JavaScript on the server.

Node.js is single-threaded but asynchronous.

Learning Progress:

Different useful built-in modules:

http, fs, URL, formidable, nodemailer

Node.js NPM

Node.js events

Node.js MySQL

Core tech: Query -- Use SQL statements to read from (or write to) a MySQL database. This is also called "to query" the database.

Useful NodeJS tips:

Heroku Node.js Support:

https://devcenter.heroku.com/articles/nodejs-support#specifying-a-node-js-version

The new best way to update npm on windows:

<https://stackoverflow.com/questions/18412129/how-do-i-update-node-and-npm-on-windows>

**7160 Weekly Report Chang Shu 18613 Content (Jul 29 - Aug 4)**

Different ways to embad a video:

Youtube

<iframe width="560" height="315" src="https://www.youtube.com/embed/YeTMN3ewl1I" frameborder="0" allowfullscreen></iframe>

youku

<iframe height=498 width=510 src='http://player.youku.com/embed/XMTcwNTIwNjU1Mg==' frameborder=0 'allowfullscreen'></iframe>

Bootstrap:

<div class="container">

<h2>Responsive Embed</h2>

<p>Create a responsive video and scale it nicely to the parent element.</p>

<h2>Aspect ratio 16:9</h2>

<div class="embed-responsive embed-responsive-16by9">

<iframe class="embed-responsive-item" src="https://www.youtube.com/embed/XGSy3\_Czz8k"></iframe>

</div>

</div>

A study of meta tag:

<meta name="viewport" content="width=device-width,minimum-scale=1.0,maximum-scale=1.0"/>

A complete tutorical of <meta>

http://blog.csdn.net/mr\_lp/article/details/53607087

http://blog.csdn.net/u011100687/article/details/48767807

Cheatsheet: HTML Codes Table

https://github.com/yangshun/tech-interview-handbook/blob/master/front-end/interview-questions.md

**7160 Weekly Report Chang Shu 18613 Content (Aug 5 - Aug 11)**

**Core: Use Heroku to demonstrate my whole work as a park of my personal website**

1. My Heroku project “charles’ Alien world” is created on the branch “heroku” of the origin

2. Use “git pull origin heroku” to download the newest edition

3. “Heroku local web” to run this app locally.

4. Use “gulp watch” to automatically adjust all stylesheet code.

5. Use “Heroku login” to use all account-related functions. (Internet access needed)

6. Use this app as a demonstration / presentation

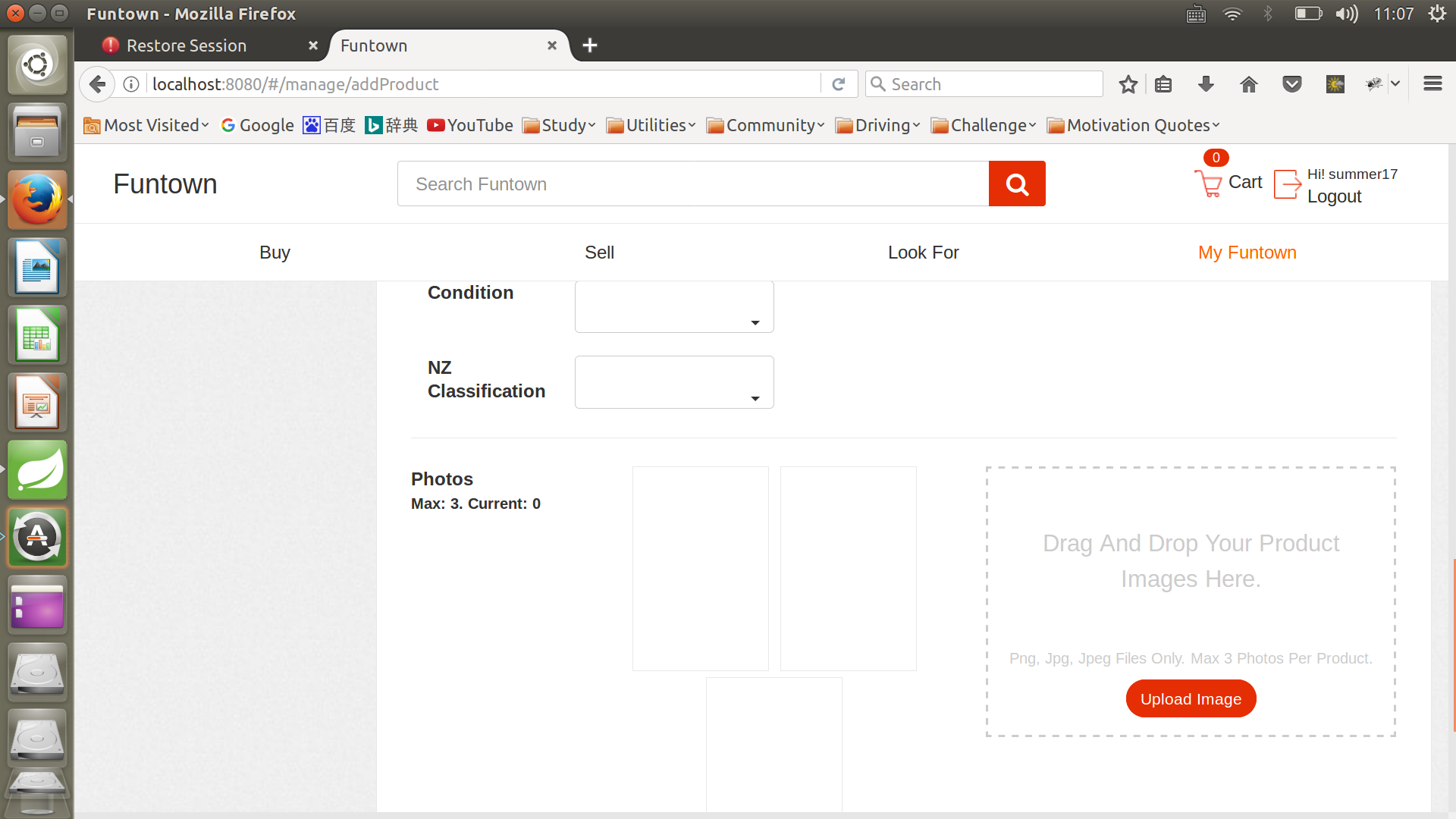
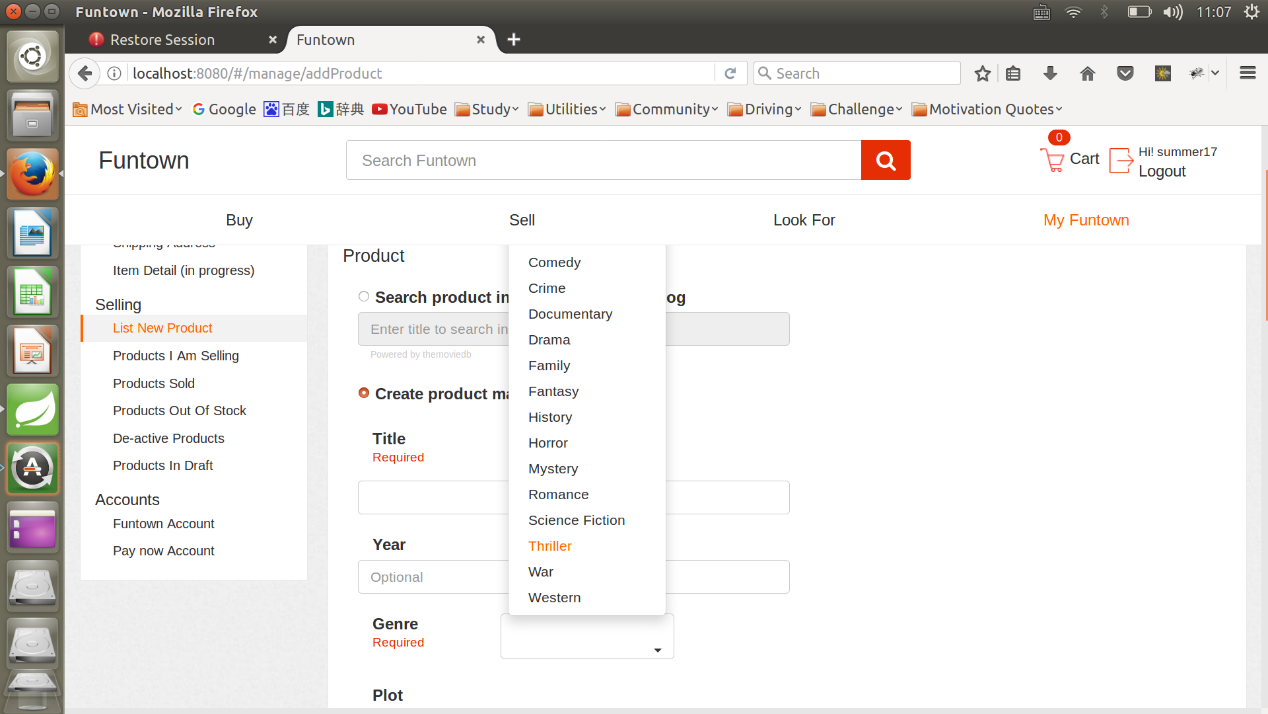
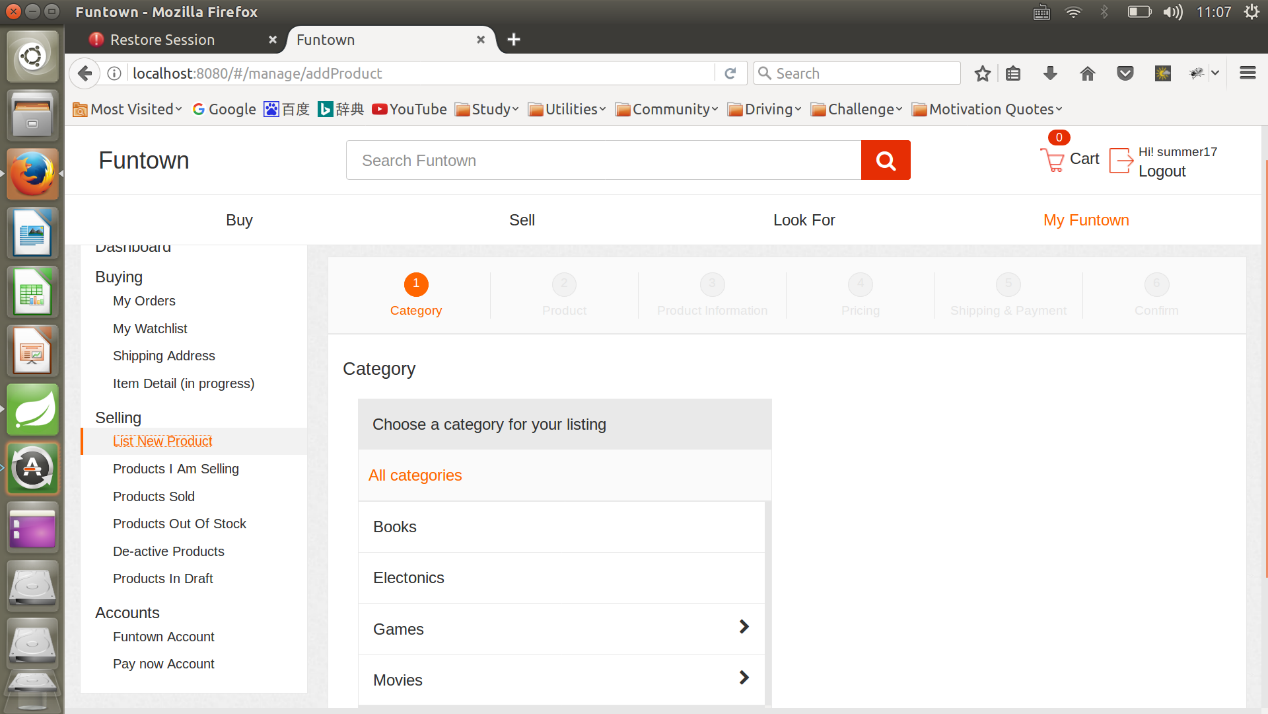
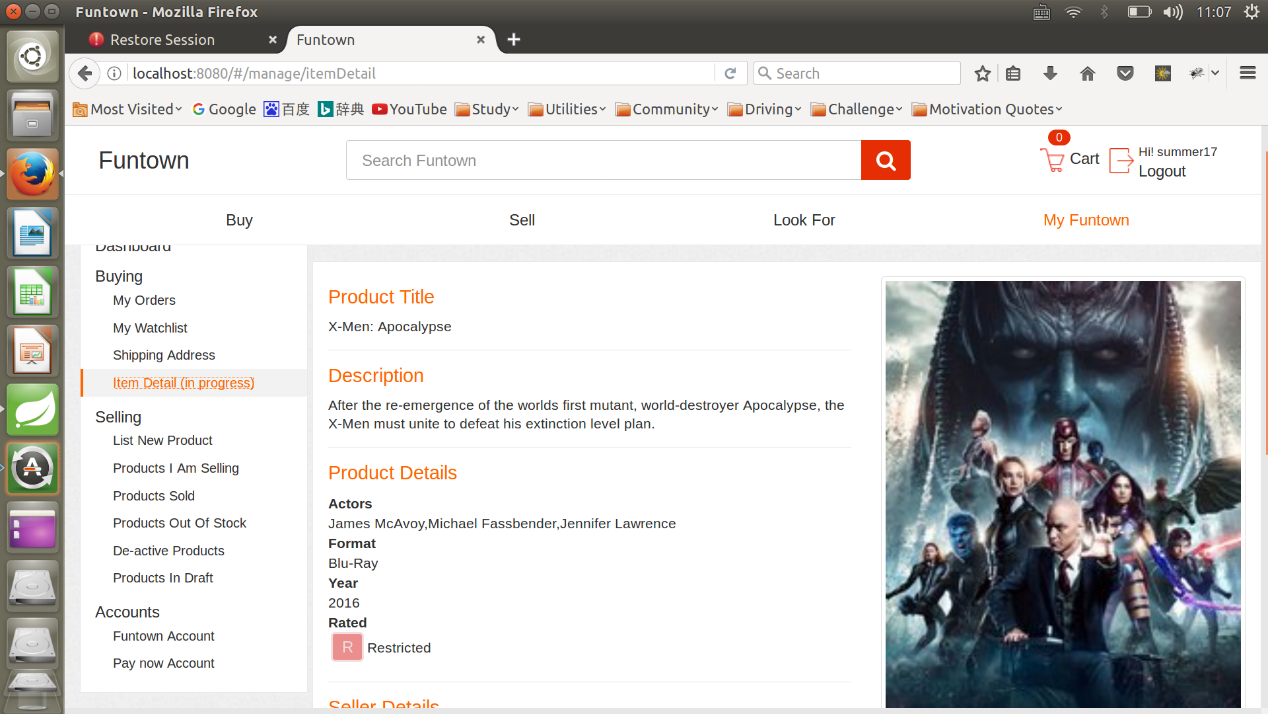
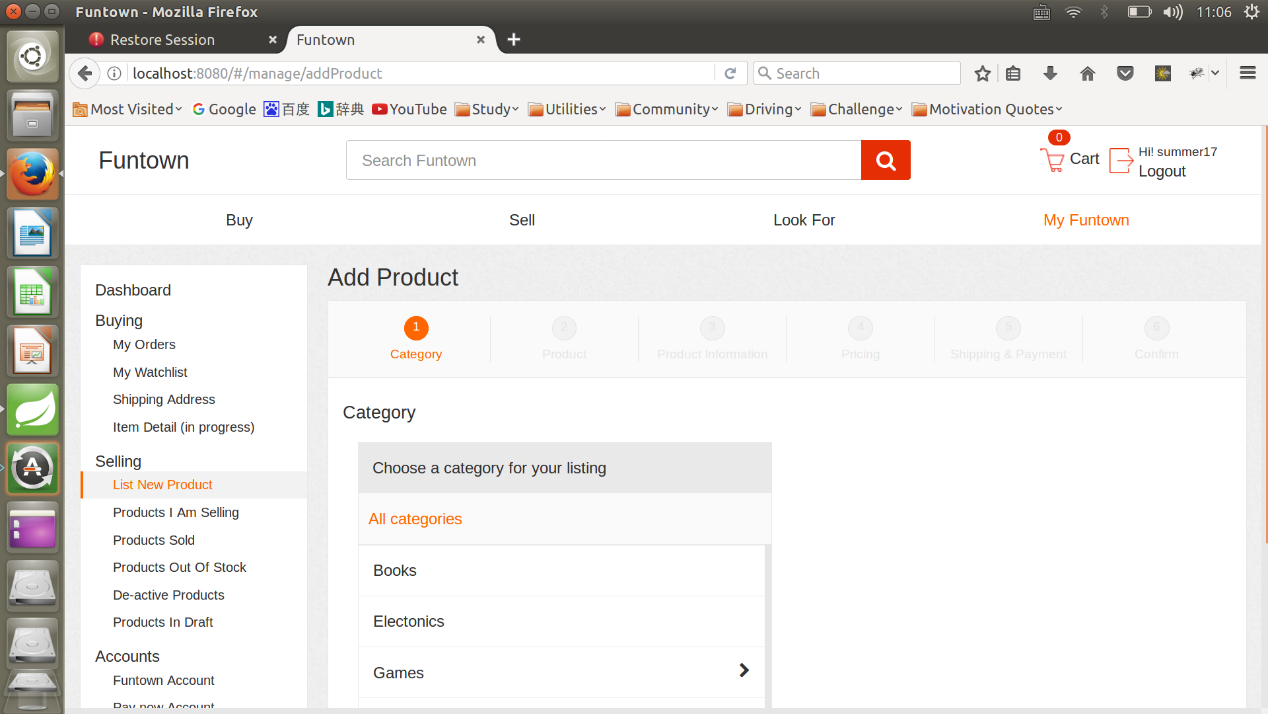
4.2 Project Final outcome

After 2 months’ effort, I made such major achievement:

1. firstly, I abandon the windows system and install Linux on my laptop and desktop. Using muti-Operation system to get maximized benefit.

2. A Complete setting up for development environment: Sublime, Apache, Spring Tool Suite™ (STS), Git, etc

3. The whole building of buying module: search function, item details function, data transferring function, etc:



4. Integrate other teammates’ code using Git

4.3 Project Critical factors

1. The suggestion and help from team leader, Aaron

2. Good communication and teamwork with team members

3. A general guideline from senior developer Kedar

4. Full devotion and focus in coding

4.4 Presentation

I made a personal website using the skills learnt from this intern. The presentation of this review is one of the main part of my project. The URL is as below:

 <https://charles-alienworld.herokuapp.com/>