

Online Shopping System

Project Documentation

Saeid Moradi

OST eProject
Aptech Education Center

IIDCO Center, IRAN
Feb, 2007

Table of Contents

TABLE OF CONTENTS.....	2
TABLE OF FIGURES.....	3
1 INTRODUCTION.....	4
2 ACKNOWLEDGMENTS.....	5
3 PROBLEM STATEMENT.....	6
3.1 ONLINE SHOPPING SYSTEM.....	6
3.2 PROJECT ASSUMPTIONS.....	6
3.3 BUSINESS RULES.....	6
4 ANALYSIS AND DESIGN.....	7
4.1 DATA MODEL ANALYSIS.....	7
4.1.1 <i>System Entities</i>	7
4.1.2 <i>ERD/Class Diagram</i>	9
4.2 USECASE MODEL.....	10
4.2.1 <i>Login Usecase</i>	10
4.2.2 <i>Register Usecase</i>	10
4.2.3 <i>Maintenance Usecase</i>	10
4.2.4 <i>Browse Usecase</i>	11
4.2.5 <i>Order Usecase</i>	11
4.2.6 <i>Usecase Model Diagram</i>	12
5 DEVELOPER'S GUIDE.....	13
5.1 IMPORT PROJECT INTO ECLIPSE.....	13
5.2 DATABASE AND CONFIGURATION.....	16
5.3 NAVIGATION PAGES.....	17
5.4 FORM PAGES.....	18
5.5 UPDATE PAGES.....	19
5.6 LIST PAGES.....	20
5.7 CONTROLLER CLASSES.....	21
6 INSTALLATION GUIDE.....	22
7 USER'S MANUAL.....	23
7.1 FOREWORD.....	23
7.2 A QUICK START.....	23
7.3 PAGE FLOW OF THE APPLICATION.....	26
7.5 TROUBLESHOOTING AND FAQ.....	33

Table of Figures

Figure 1 – ERD Diagram.....	9
Figure 2 – Usecase Model Diagram.....	12
Figure 3 – Import Part1.....	13
Figure 4 – Import Part2.....	14
Figure 5 – Import Part3.....	15
Figure 6 – Import Part4.....	15
Figure 7 – Database And Configuration.....	16
Figure 8 – Navigation pages.....	17
Figure 9 – Form pages.....	18
Figure 10 - Update pages.....	19
Figure 11 – list pages.....	20
Figure 12 – Controller Classes.....	21
Figure 13 – Login Page.....	23
Figure 14 – Change Admin Password.....	24
Figure 15 – Update Product Info.....	25
Figure 16 – Home Page.....	26
Figure 17 – Browse Page.....	27
Figure 18 – Shopping Basket.....	28
Figure 19 – Confirm Order.....	29
Figure 20 – Registration Page.....	30
Figure 21 – After Registration.....	31
Figure 22 – Order History.....	32
Figure 23 – Error log in project.....	33

1 Introduction

This project was our task at end of OST courses , And we had specification with eProject Group in Aptech Education Center . In another way , we complete all of OST courses in second year , And now we must provide one project with use of what we learn during these courses.

Our Time for doing this project was 12 days and we start from 24-Jan-2007 until 06-Feb-2007 . During this time we had some session with our faculty In Aptech school , And he guide us in very topics which we were face with them, And every body was response of his/her project for develop. My eProject was about online shopping system , and this documataion describe my project.

2 Acknowledgments

As with any documataion , i like to acknowledge those who help me to do this project.

A big thank to Mr.Hooshmandi Rad our faculty in eProject course in Aptech Education Center

He help us to increase our speed in all Phases of project especially Development Phase.

A special thank to Mr.Nonakht another faculty ,that help me to have better Documentation phase.

A huge thank to Internet people and Open Source Community .

Last but not least , I'd like to thank my family for their continuous supports, especially my mother for her delicious foods .

3 Problem Statement

3.1 Online Shopping System

In This project we have one Stakeholder by name jenny , jenny want have one website for selling imitation jewellery online 24 hours , website must have ability add categories and product and update information of each product ,Also she want have backup system for database and have two report from top 10 customer and product that had most buying and selling .

Customers must have registration page for login into system , ability to browse and search product by name and categories , adding products to the shopping basket for check out and register order.

3.2 Project Assumptions

Using from <http://adodb.sourceforge.net/> , ADOdb is a database abstraction library for PHP, ADOdb currently supports many databases like *MySQL, PostgreSQL, Interbase, Oracle, Ms SQL, Foxpro, Access, DB2, LDAP, ...*

Architcture of project is flexible for adding another features in future , OOP aspect in our project easily help us to add new entity or change existing entity.

This project is Validate by www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd standard Validation for HTML Pages .(If your browser be FireFox And You have Extension HTML Validator, you can see easily)

3.3 Business Rules

In many page of project we are face by registration forms or update forms or list of one entity , In all cases we have one behavior in our action but entity of form or list are different .

This topic can help us for have better understaning in business rules , architcture and navigation.

I use this slouation in my project for this subject:

Use of controller for each entity help me to have a switcher which guide entity to correct page .

Use of controller will separate data layer from ui layer , and help us to focus in development each layer of project separately form others layers.

4 Analysis and Design

4.1 Data Model Analysis

4.1.1 System Entities

4.1.1.1- Users

This entity contains same Information of each user in our system , For example we have user by role customer that have Username and Password, these fields are same by another users by different role , So we take all common fields in users entity

- USERNAME varchar(40)
- PASSWORD varchar(64)
- NAME varchar(40)

4.1.1.2- Roles

This entity contains all roles that we can have in our project , by this entity we can easily add new role in our roles . For example now our project have two main role 1_ROLE_ADMIN 2_ROLE_CUSTOMER but we can add new role like ROLE_MANAGER easily, that have different Permission from other roles.

- ID varchar(40)
- NAME varchar(40)

4.1.1.3- Category

Products in our project are varied , and we need category entity to arrange them into different category. By categorise , we can easily have browse between products.

- ID int(11)
- NAME varchar(40)

4.1.1.4- product

This entity is for store information of each goods that we want sell in our website.

- ID int(11)
- NAME varchar(40)
- DESCRIPTION varchar(255)
- PRICE decimal
- PICTURE mediumblob

4.1.1.5- Customer Type

This is one entity for categorise customer into different types , for example family, friend , co-worker, ...
This entity is like category entity for product entity.

- ID smallint
- NAME varchar(40)

4.1.1.6- Customer

In This Entity we have Information of User That register as customer in our website , customer fill all fields about his or her identify , And system automatically assign role customer to him or her .

- UID varchar(40)
- ADDRESS varchar(255)
- EMAIL varchar(100)
- PHONE varchar(20)
- CELL varchar(20)
- BIRTHDAY datetime
- TYPE smallint

4.1.1.7- Orders

In this entity we store informations, which is important in selling transaction and don't have repeat.
For example UID of customer or PDATE times when order was paid , have one value , but PID product id can have more than value , these important information will be to store in order details entity.

- ID int(11)
- PDATE datetime
- TOTAL decimal
- UID varchar(40)

4.1.1.8- Order Details

In this entity we have additional information of selling transaction , this information may have repeat in rows of data but each row is unique by OID(orderID) and PID(productID) with each other.

- OID int(11)
- PID int(11)
- QUANTITY smallint
- PRICE decimal

4.1.2 ERD Diagram

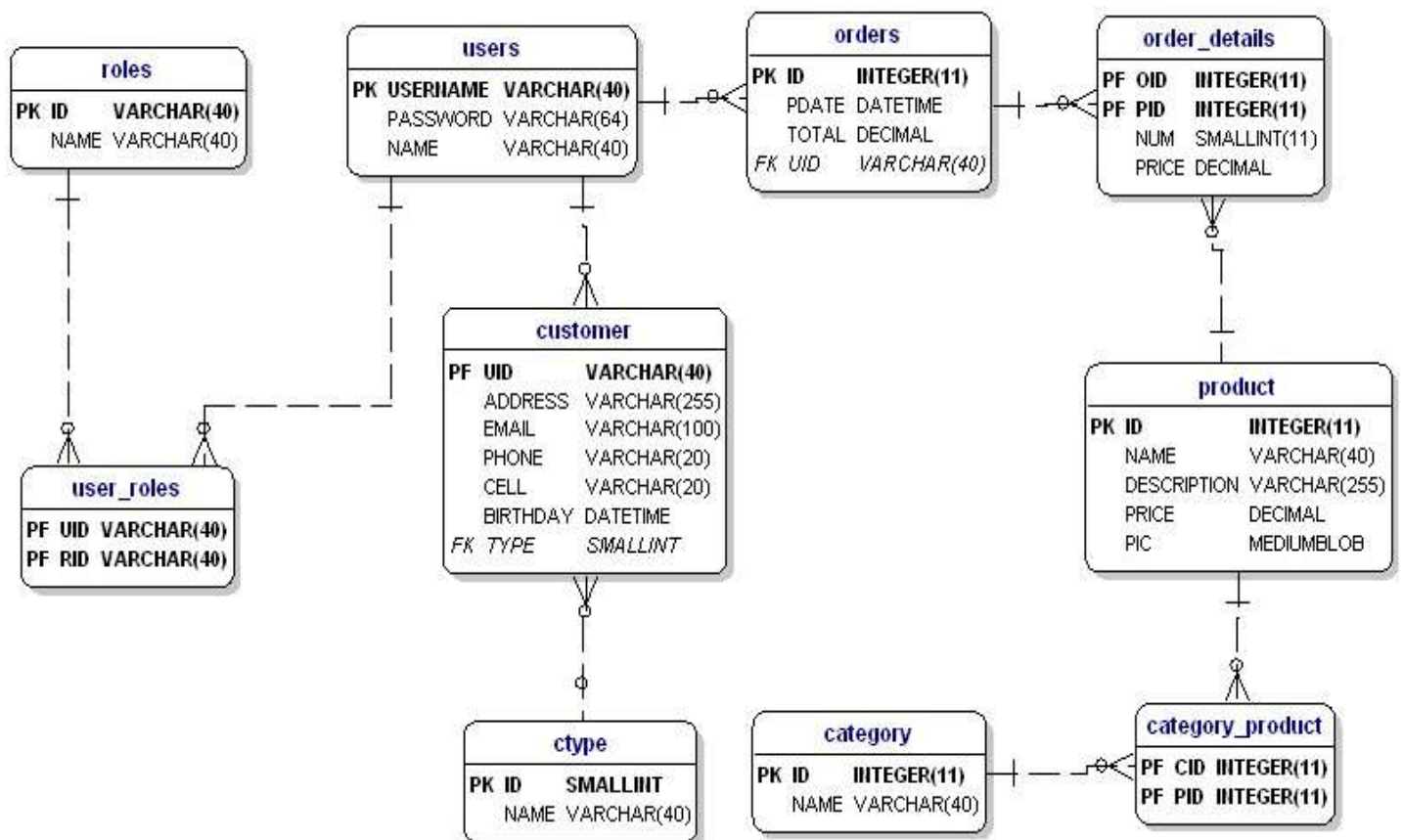


Figure 1 – ERD Diagram

4.2 Usecase Model

4.2.1 Login Usecase

Usecase Features	Descriptions
Usecase Name	Login
Usecase Actor	Admin , Customer
Usecase Preconditions	Registration Must be doing before this Usecase.
Usecase Description	This Usecase Describe How User Log into system.
Usecase Results	User log into system and have more ability and functionality related to role of his or her .

4.2.2 Register Usecase

Usecase Features	Descriptions
Usecase Name	Register
Usecase Actor	Admin, Customer
Usecase Preconditions	No preconditions
Usecase Description	This Usecase shows customer registration phase.
Usecase Results	Customer Can Login Into system

4.2.3 Maintenance Usecase

Usecase Features	Descriptions
Usecase Name	Maintenance
Usecase Actor	Admin
Usecase Preconditions	No preconditions
Usecase Description	In this usecase Admin of Website can add, delete update category and product , get reports from orders , and backup from database .
Usecase Results	All changes be effected .

4.2.4 Browse Usecase

Usecase Features	Descriptions
Usecase Name	Browse
Usecase Actor	Admin, Customer
Usecase Preconditions	No preconditions
Usecase Description	Goal of this Usecase is provide easy way for customer to browse product by category and name customer can add products into shopping cart.
Usecase Results	Customer added some product into shopping cart.

4.2.5 Order Usecase

Usecase Features	Descriptions
Usecase Name	Order
Usecase Actor	Admin, Customer
Usecase Preconditions	Customer Must Be Log into system.
Usecase Description	This Usecase started from when customer click on the shopping basket and over when confirm order.
Usecase Results	Customer have history of his/her order. Admin have history of all orders by all customers.

4.2.6 Usecase Model Diagram

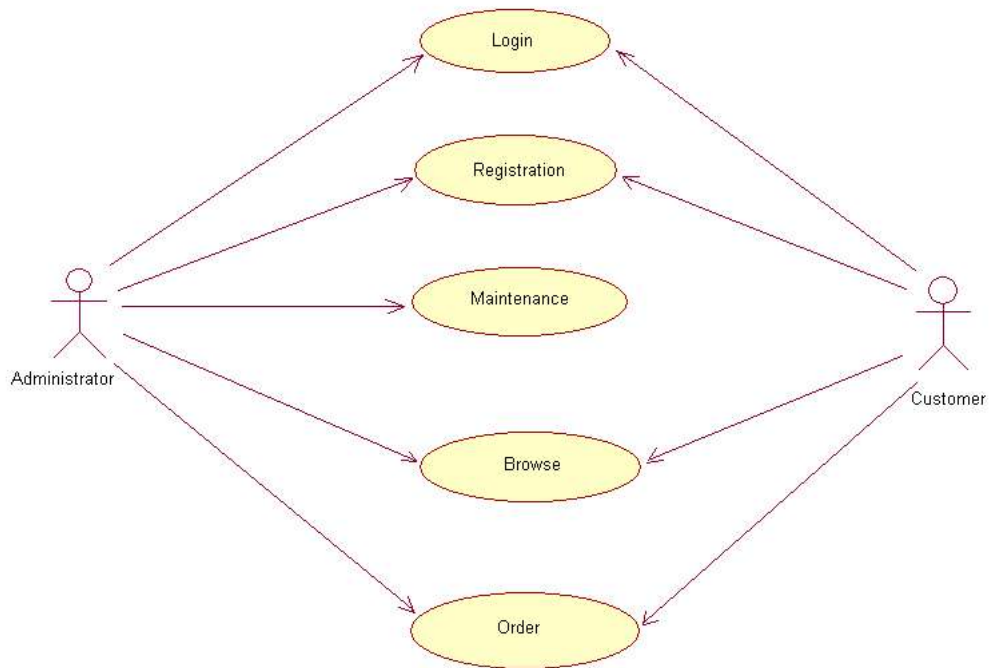


Figure 2 – Usecase Model Diagram

5 Developer's Guide

In this part i must show how one developer can consider my project or have development .

After extarct zip file of project (saeid_moradi_eproject-1.0.zip) , you're face with one file and one folder , the file(project_report.pdf) is this document , And Folder (www.jijws.com) is source of our application.

www.jijws.com is root folder of our project And we must copy this folder into document root of our Apache Server. After Copy You Need one text editor for edit and see files . Your editor can be every things or any IDE , But I will show you how work on this source with eclipse IDE .

5.1 Import project into eclipse

As you see in below you must click on File > Import .

Also pay attention into php perspective must be selected.

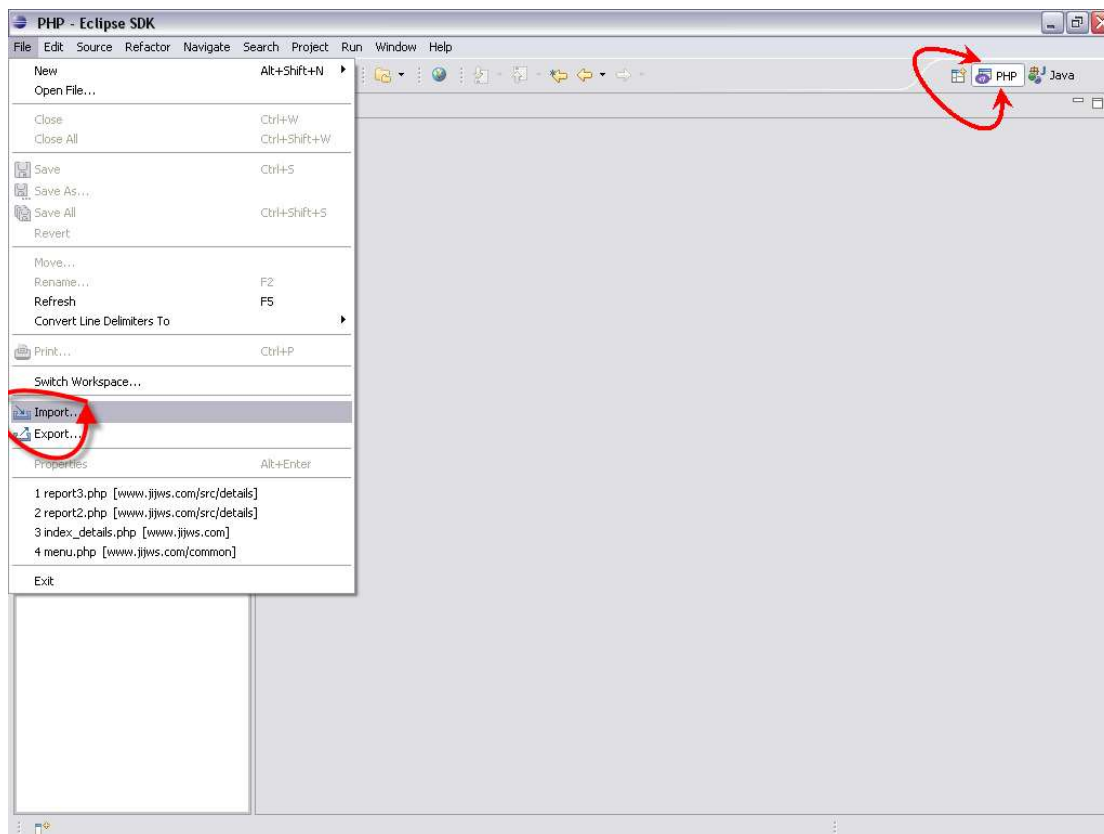


Figure 3 -Import part1

Choose Existing Projects into Workspace And Click Next.

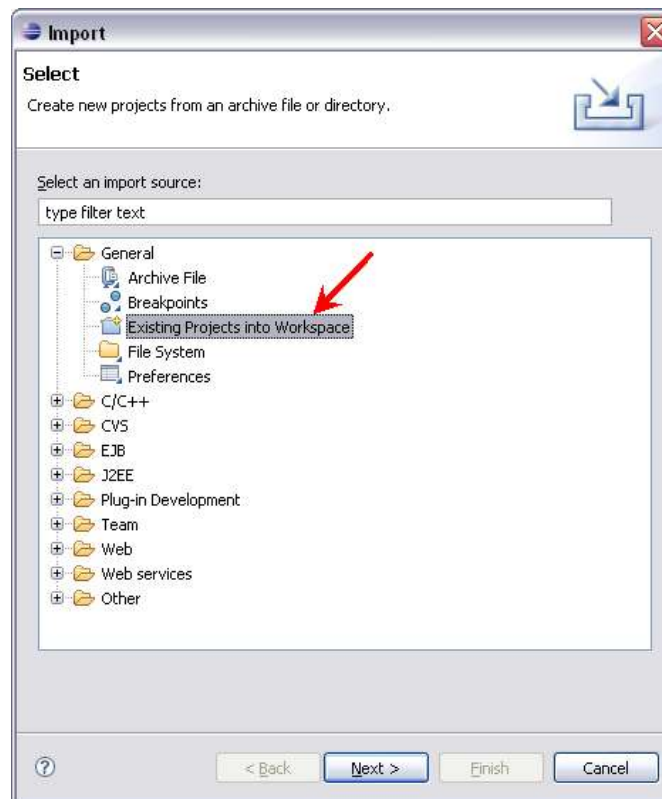


Figure 4 – Import part2

In this phase you must do these tree step

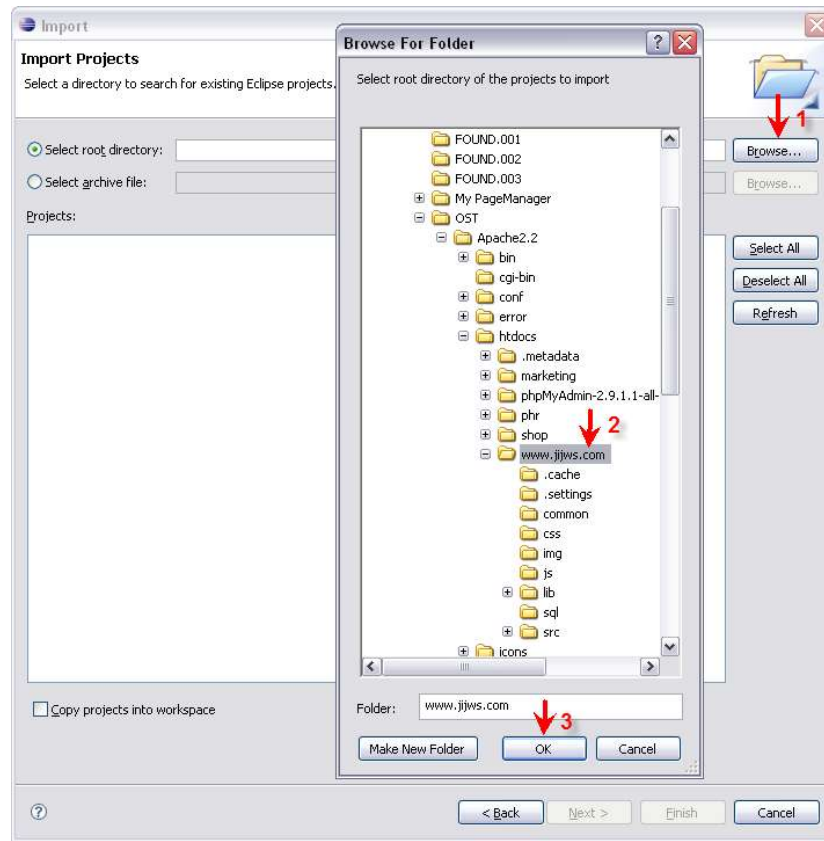


Figure 5 – import part3

and at last just click finish to complete import phase .

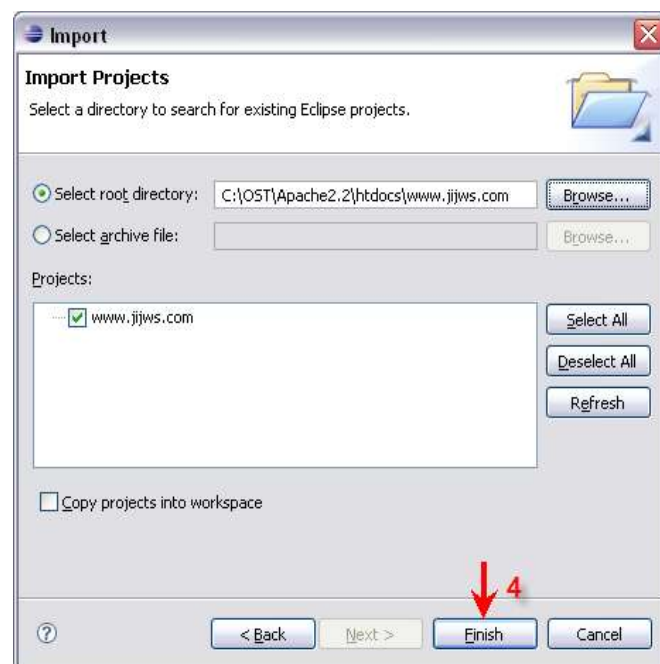


Figure 6 –import part4

5.2 Database and configuration

In below figure we have 3 part and i explain you each part briefly.

Part1 is about one file by name db.sql , by this file we can import all database setting in our mysql database , you can do this work by phpmyadmin or run this script with mysql command line.

Part2 is about database setting and from top into down we have hostname, username, password, database name , and my database name is jijws0003 in this case you can change that by each you set in db.sql file.

Part3 depends on your Operating System , and don't forget last / .

That's it , Now You can go into below address by your browser.

<http://localhost/www.jijws.com/>

And By below username and password You can login into system by role admin

username = admin

password = admin

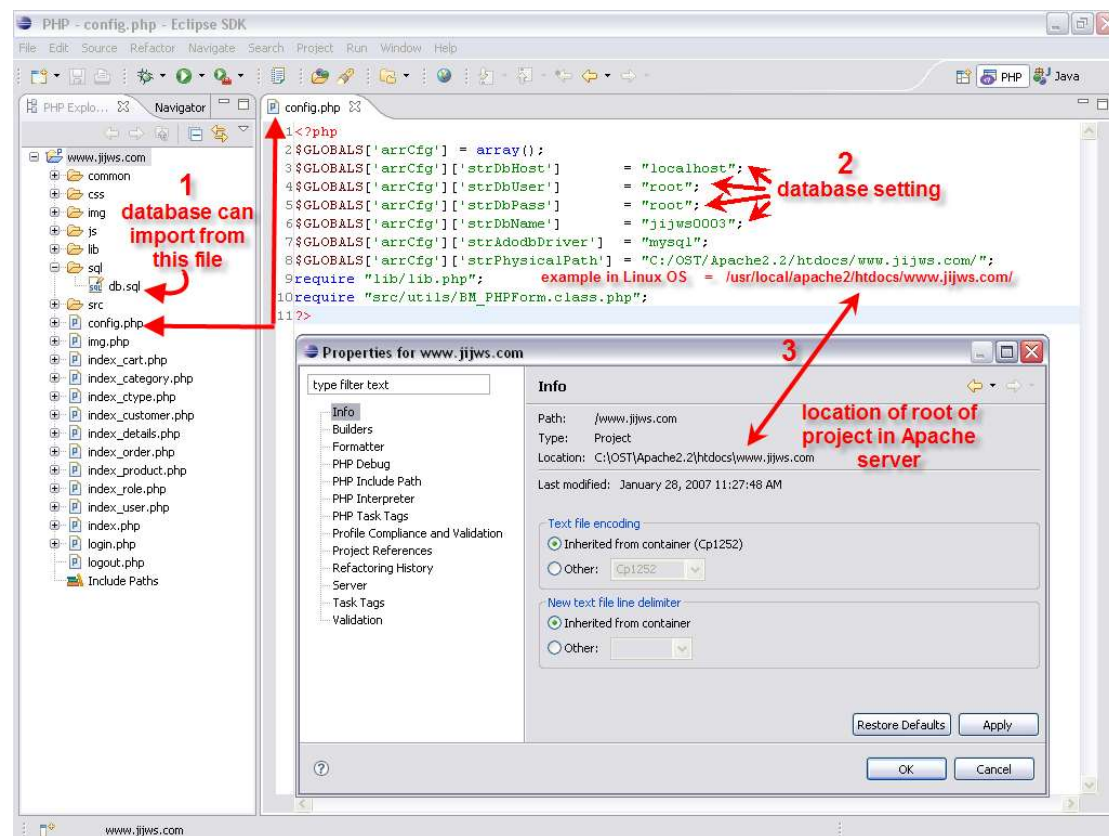


Figure 7 – database and configuration

5.3 Navigation pages

In below figure , we consider navigation of page for one entity , this navigation is same for another entities .

Our entity in this figure is user, that need admin permission for edit and list .

Scenario start from here , when admin click User management on the menu .

By click we goes into this URL: http://localhost/www.jijws.com/index_user.php?action=new

In first click we have our action in \$_REQUEST ,but we must take action in our form by hidden field in our \$_POST , you can see this part As No. 1

No.2 Is next step in our senario because action was = "new" .

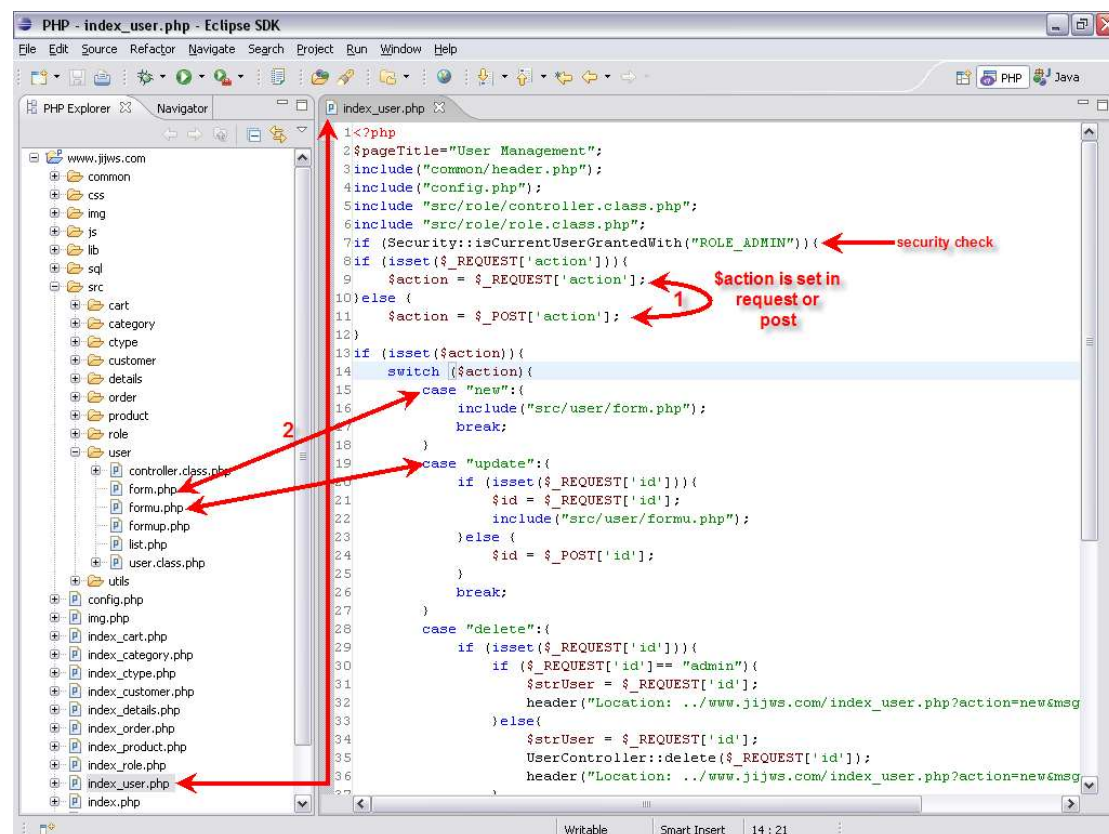


Figure 8 – Navigation pages

5.4 form pages

This file(/src/user/form.php) is for create new user into system by assign new Role .

As you can see in below figure , we have one object from class BM_PHPForm()
(/src/utlis/BM_PHPForm.class.php) , this object help us to have felaxible and neat form .

This class have some methods and fields can easily help us to provide validation, show messages and provide css design easily ,and take care of value each field in every step of validation.
And also is wonderful help for have neat source code.

```
<?php
$objUserForm = new BM_PHPForm();
$objUserForm->UseLocator();
$objUserForm->SetIndent(" ");
$arrRoles = RoleController::findAll();

if (isset($_REQUEST['msg']) && $_REQUEST['msg'] == "err"){
    $strUser = $_REQUEST['user'];
    $objUserForm->SetError("delete","User [$strUser] Cann't Be Deleted .");
}
if (isset($_REQUEST['msg']) && $_REQUEST['msg'] == "inf"){
    $strUser = $_REQUEST['user'];
    $objUserForm->SetInfor("delete","User [$strUser] Deleted Successfully!");
}

if ($objUserForm->IsPost()){
    $entUser0 = UserController::find($objUserForm->arrPostData['username']);
    $objUserForm->ChEmptiness("username", "Username is empty.");
    $objUserForm->ChWithRegex("username", "Username Have Invalid Chars.", "/^([a-zA-Z0-9_\\.\\-\\~])*$/");
    $objUserForm->ChMinLength("username", "Username must not be less than 4 chars", 4);
    $objUserForm->ChMaxLength("username", "Username must not be more than 40 chars.", 40);
    $objUserForm->ChCustomVal("username", "Username Exist ! Please Choose Another .", $entUser0->username == "guest");
    $objUserForm->ChEmptiness("name", "Name is empty.");
    $objUserForm->ChMaxLength("name", "Name must not be more than 40 chars.", 40);
    $objUserForm->ChWithRegex("name", "Name Have Invalid Chars.", "/^([a-zA-Z0-9_\\.\\-\\~\\s])*$/");
    $objUserForm->ChEmptiness("password", "Password is empty.");
    $objUserForm->ChMinLength("password", "Password must not be less than 4 chars", 4);
    $objUserForm->ChMaxLength("password", "Password must not be more than 12 chars.", 12);
    $objUserForm->ChCustomVal("password", "Passwords must be same.", $objUserForm->arrPostData['password'] == $objUserForm->arrPostData['password']);

    if ($objUserForm->IsValid()){
        $entUser = new User($objUserForm->arrPostData['username'], md5($objUserForm->arrPostData['password']), $objUserForm->arrPostData['password']);
        UserController::save($entUser);
        $uroles = array();
        foreach($arrRoles as $role){
            if (isset($objUserForm->arrPostData[$role->id])){
                $uroles[] = $role->id;
            }
        }
        if (!sizeof($uroles) == 0){
            UserController::saveRoles($entUser, $uroles);
        }
        $objUserForm->SetInfor("submit", "User Added Successfully !");
        header("Location: ../www.jijws.com/index_user.php?action=update&id=$entUser->username&msg=add");
    }
}

print $objUserForm->ErrMsg();
print $objUserForm->InfMsgs();

print $objUserForm->Open("userform", "User Registration Form");
print $objUserForm->HiddenField("action", $action);
print $objUserForm->TextBox("username", "Username:", "");
print $objUserForm->TextBox("name", "Name:", "");
print $objUserForm->TextBox("password", "Password:", "", array(), "password");
print $objUserForm->TextBox("password2", "Confirm Password:", "", array(), "password");
print " <div style='padding-left: 100px; width: 30px;'>Roles:</div>\r\n";
foreach ($arrRoles as $role){
    print $objUserForm->CheckBox("$role->id", $role->name, array(), false, $role->id);
}
print $objUserForm->SubmitButton("submit", "Register", array(), "submit");
//print $objUserForm->Reset("reset", "Reset", array(), "reset");

print $objUserForm->Close();
?>
```

Figure 9 – form pages

5.5 Update pages

This file(/src/user/formu.php) is for update of user entity .

As you can see in below figure , in top of page we take information of entity from database , and then fill value of each fields with these values of entity.

We just need right value for each field in first time which we came into update pages , so after this step our form object will do this work for us automatically.

```

<?php
$objUserForm = new BM_PHPForm();
$objUserForm->UseLocator();
$objUserForm->SetIndent("    ");

if (isset($_REQUEST['msg']) && $_REQUEST['msg'] == "add"){
    $objUserForm->SetInfor("useradd", "User Added Successfully !");
}

$entUser = UserController::find($id);
$entUser = UserController::findUserRoles($entUser);
$arrRoles = RoleController::findAll();

if ($objUserForm->IsPost()){
    $objUserForm->ChEmptiness("name", "Name is empty.");
    $objUserForm->ChMaxLength("name", "Name must not be more than 40 chars.", 40);
    $objUserForm->ChWithRegex("name", "Name Have Invalid Chars.", "/^[a-zA-Z0-9_\\.\\-\\s]*$/");

    if ($objUserForm->IsValid()){
        $entUser = new User($entUser->username, "", $objUserForm->arrPostData['name']);
        UserController::update($entUser);
        UserController::deleteRoles($entUser->username);
        $uroles = array();
        foreach($arrRoles as $role){
            if(isset($objUserForm->arrPostData[$role->id])){
                $uroles[] = $role->id;
            }
        }
        if (sizeof($uroles) == 0){
            UserController::saveRoles($entUser, $uroles);
        }
        $objUserForm->SetInfor("submit", "User Updated Successfully !");
    }
}

print $objUserForm->ErrMsgs();
print $objUserForm->InfMsgs();

print $objUserForm->Open("userform", "User Update Form");
print $objUserForm->HiddenField("action", "action");
print $objUserForm->HiddenField("id", "id");
print $objUserForm->TextBox("username", "Username:", $entUser->username, array('readonly' => 'readonly'));
print $objUserForm->TextBox("name", "Name:", $entUser->name);
print "    <div style='padding-left: 78px; padding-bottom: 10px;'>Password: <a href='\"../www.jijws.com/index_user.pl\"'>
print "    <div style='padding-left: 102px;'>Roles:</div>";
foreach ($arrRoles as $role){
    if (in_array($role->id, $entUser->roles)){
        print $objUserForm->CheckBox("$role->id", $role->name, array(), true, $role->id);
    }else{
        print $objUserForm->CheckBox("$role->id", $role->name, array(), false, $role->id);
    }
}
print $objUserForm->SubmitButton("submit", "Update", array(), "submit");
print $objUserForm->Close();
?>

```

this method will find entity user by \$id
 this method will find all roles of entity user
 this method will find all roles of system
 arrPostData is one array of post data, that have been filtered to be safe by object form
 set value of the field in first view of update page
 each field can have one array for customise spacial field. And in this case, i add attribute readonly to this field.
 set checkbox checked or not checked in first view of update page.

Figure 10 – update pages

5.6 List Pages

List pages are simple and just show us list of entities.

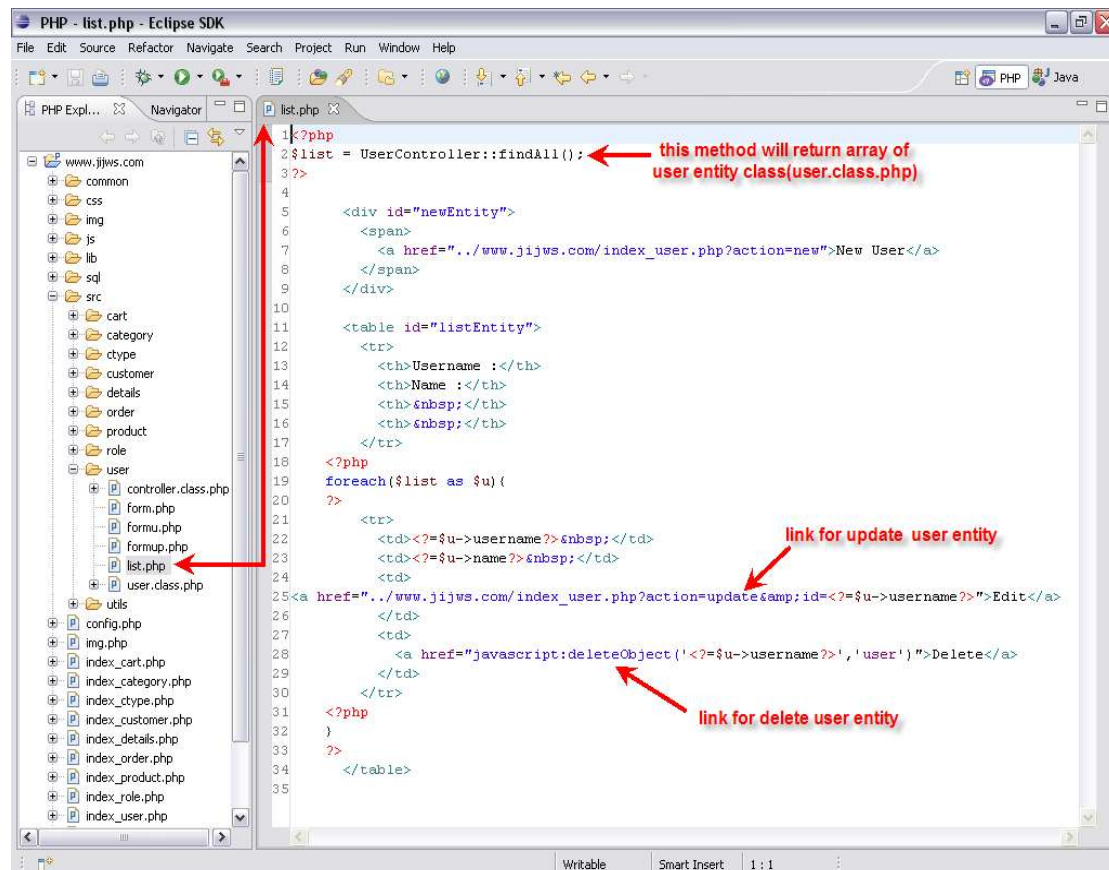


Figure 11 – list pages

5.7 Contollor classes

Controller.class.php files are our classes for connect into database and execute the queries.

We can have access in these classes from all part of project.

And in many cases we need to call one function of another entity for current entity.

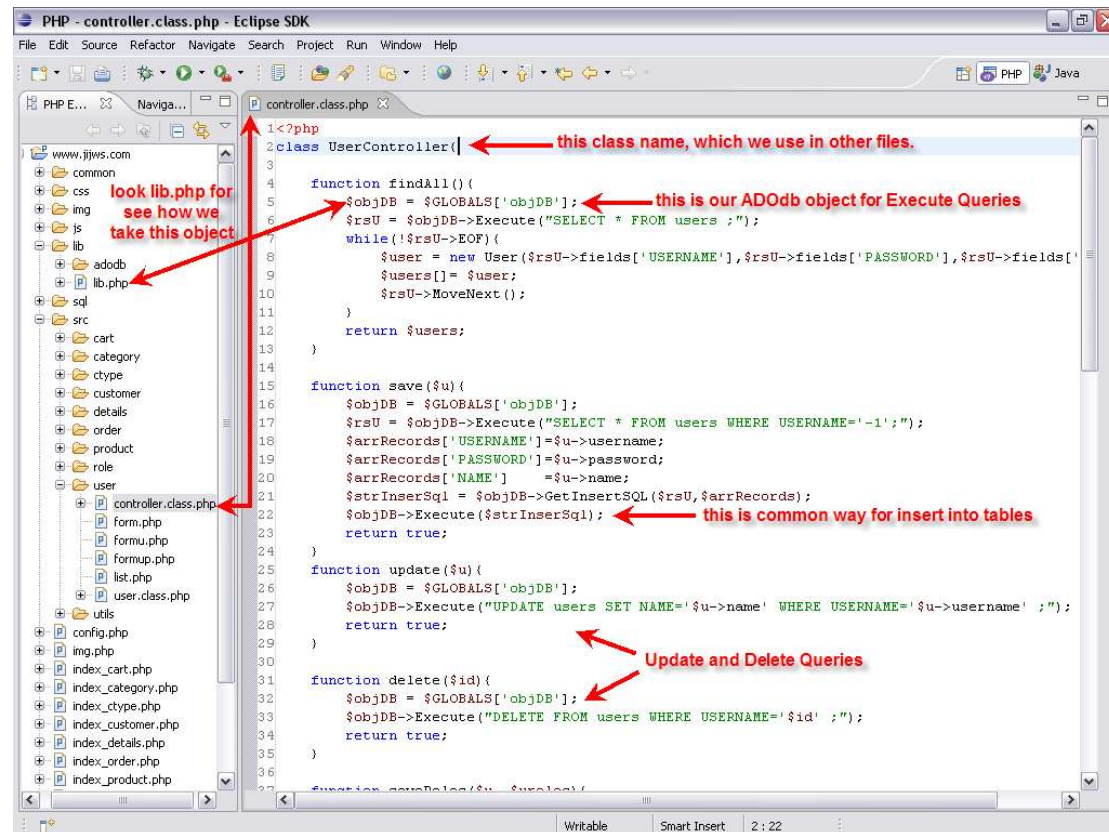


Figure 12 – controller classes

6 Installation Guide

In this section i must describe what application and environment did you need for run this application.

First i like introduce my testing enviroment for you , then i explain another ability.

In this project my Oprating System was Window xp and my Application Server was Apache-2.2.3

And my database was mySQL-5.0.20 , And my Scripting Language was PHP-5.2.0.

Windows XP

Apache -2.2.3

mySQL -5.0.20

PHP -5.2.0

But abilities , the OS can be every things by support apache server, mysql, and PHP.(e.g Linux)

In this project we don't use any features of Apache server , so version of apache isn't important.

MySQL , use from ADOdb help us to walk even bettween other databases (see ADOdb for more info).

PHP , in this case , i also try to compatible with version 4 , and my test was good with feature of eclipse!

Yes eclipse php plug-in can compile your php code with version 4 or version 5 .

About how to install this application , we had complete explanation in part 5.Developer's guide .

But here, again we have brief explain.

1_After download and extarct zip file (saeid_moradi_eproject-1.0.zip) you have one file and one folder

2_name of folder is (www.jijws.com) , copy this folder into your document root of apache server.

3_import your database setting with this script file (/www.jijws.com/sql/db.sql).

4_edit this file (/www.jijws.com/config.php) for change database setting into yours.

5_go into this URL by your browser (<http://localhost/www.jijws.com/>).

6_If you want log into system by role administartor the username=admin password=admin

7 User's Manual

7.1 Foreword

In this part we will show you how work with this application by different roles .
We have two main roles in this application , Role Administartor and Role Customer.
Administartor is one who that have most access into this application by more functionality.
Customer is another role by less access into this system and by limit functionality.

7.2 Quick Start

7.2.1 Login Page

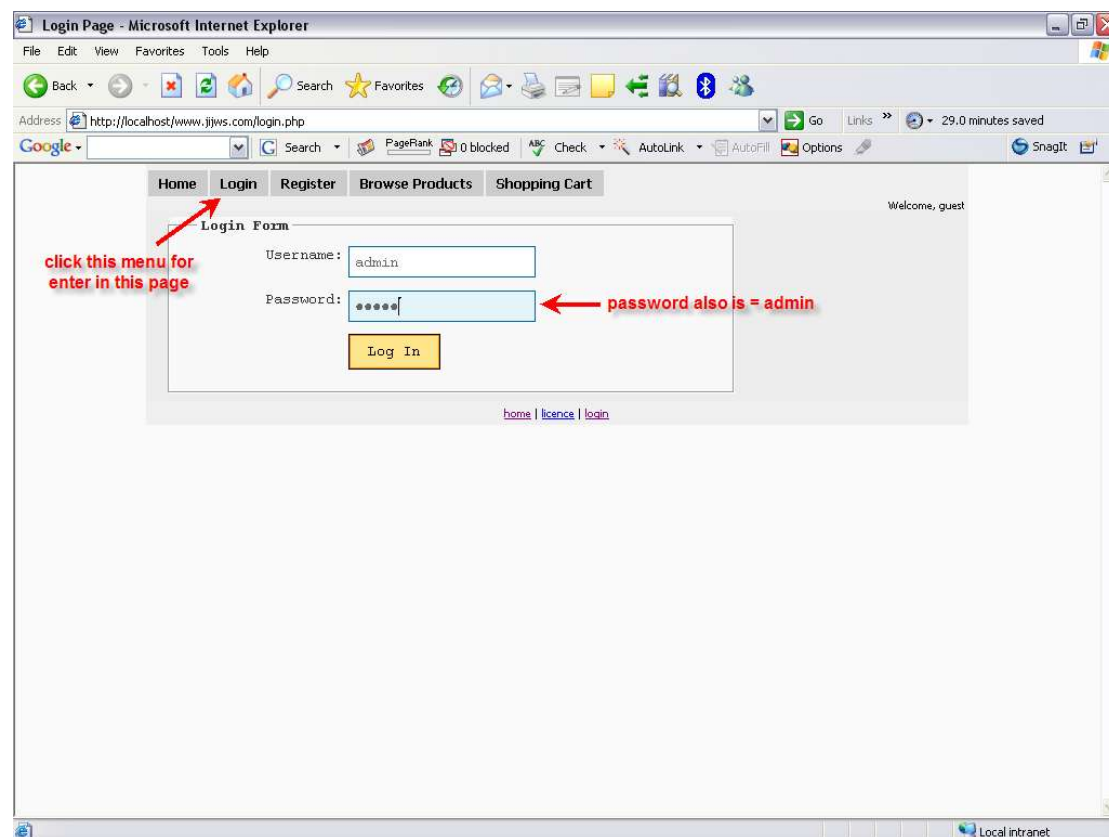


Figure 13 – Login Page

7.2.2 Change Admin Password

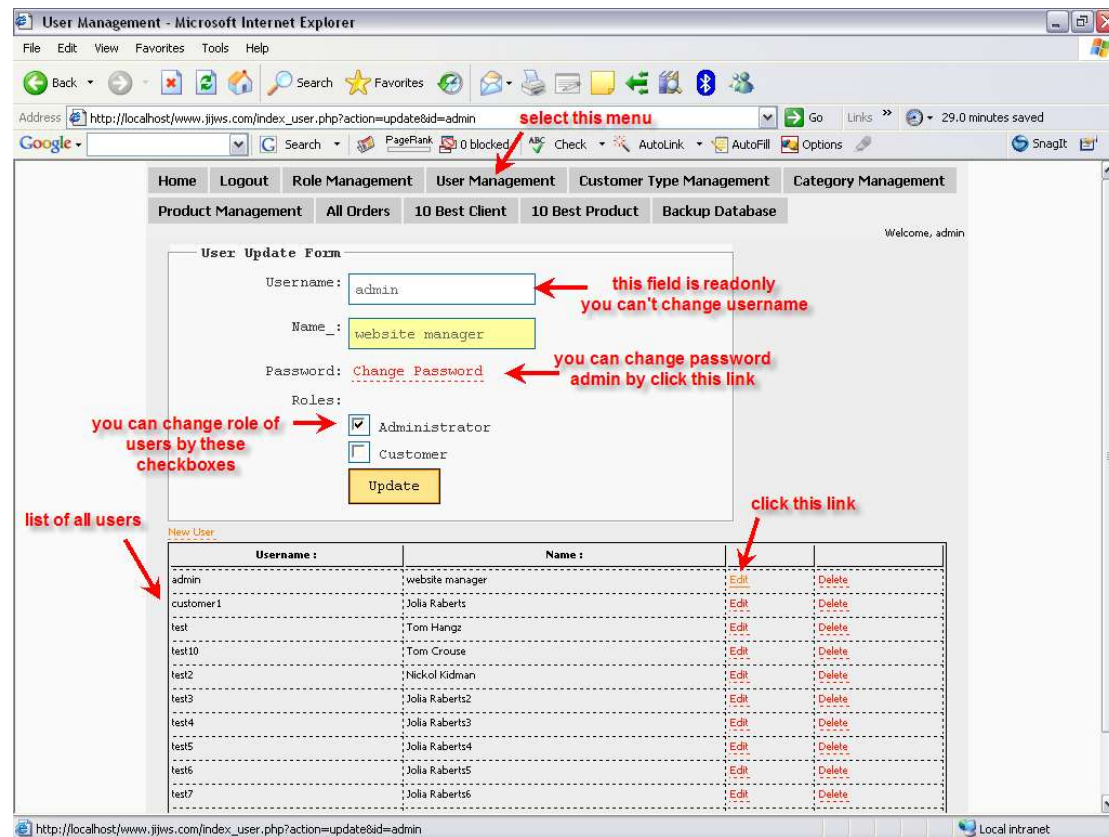


Figure 14 – Change Admin Password

7.2.3 Update Product Info

select this menu

all fields are editable

list of products

picture will be store in two place
1_database
2_/www.jijws.com/img/

ID	NAME	DESCRIPTION	PRICE	Edit	Delete
1	prod1	this is new description for production 1, last edit is 2007-02-05	1000	Edit	Delete
2	prod2	Feature: 1) Fashionable and elegant design 2) Material: Czech crystal, rhinestone, alloy 3) Competitive price and high quality 4) Customers requirement are welcomed	2000	Edit	Delete
3	prod3	prod3 description	3000	Edit	Delete
4	prod4	prod4 description	3000	Edit	Delete
5	prod5	prod5 description	3000	Edit	Delete
6	prod6	prod6 description	3000	Edit	Delete
7	prod7	prod7 description	3000	Edit	Delete

Figure 15 – Update Product Info

7.3 Page Flow of the Application

7.3.1 Home Page

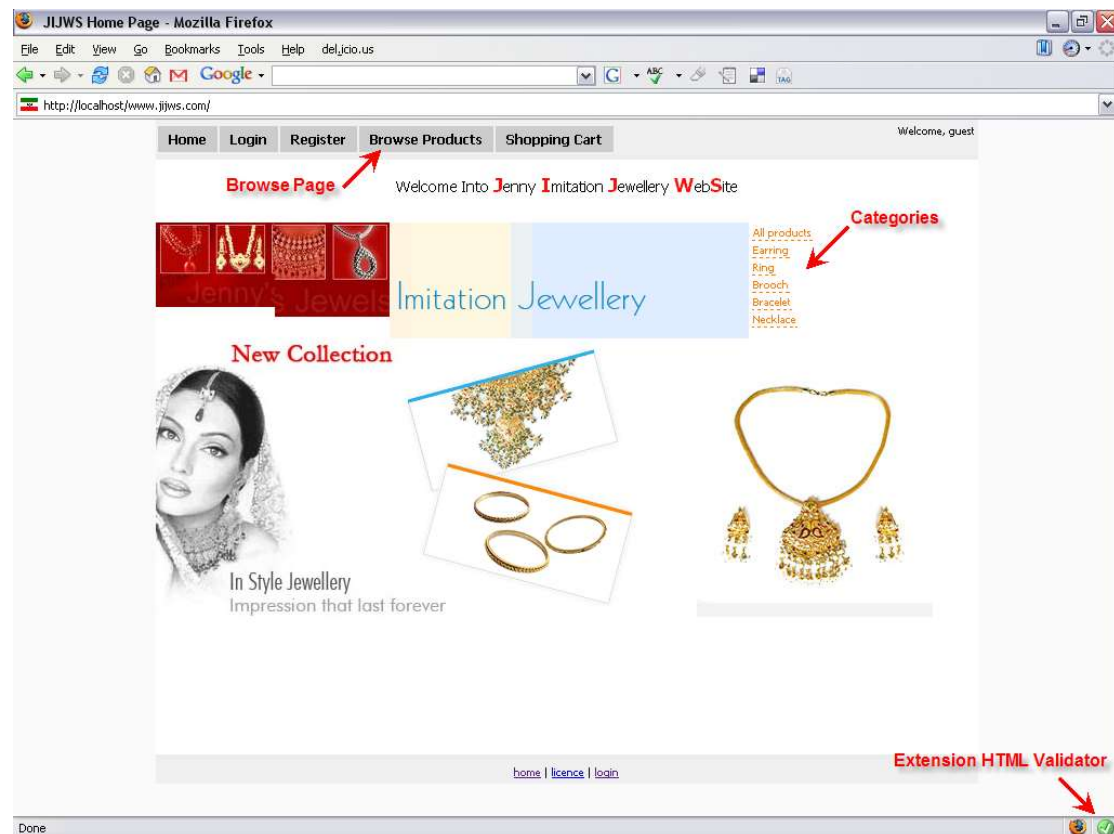


Figure 16 – Home page

7.3.2 Browse Page

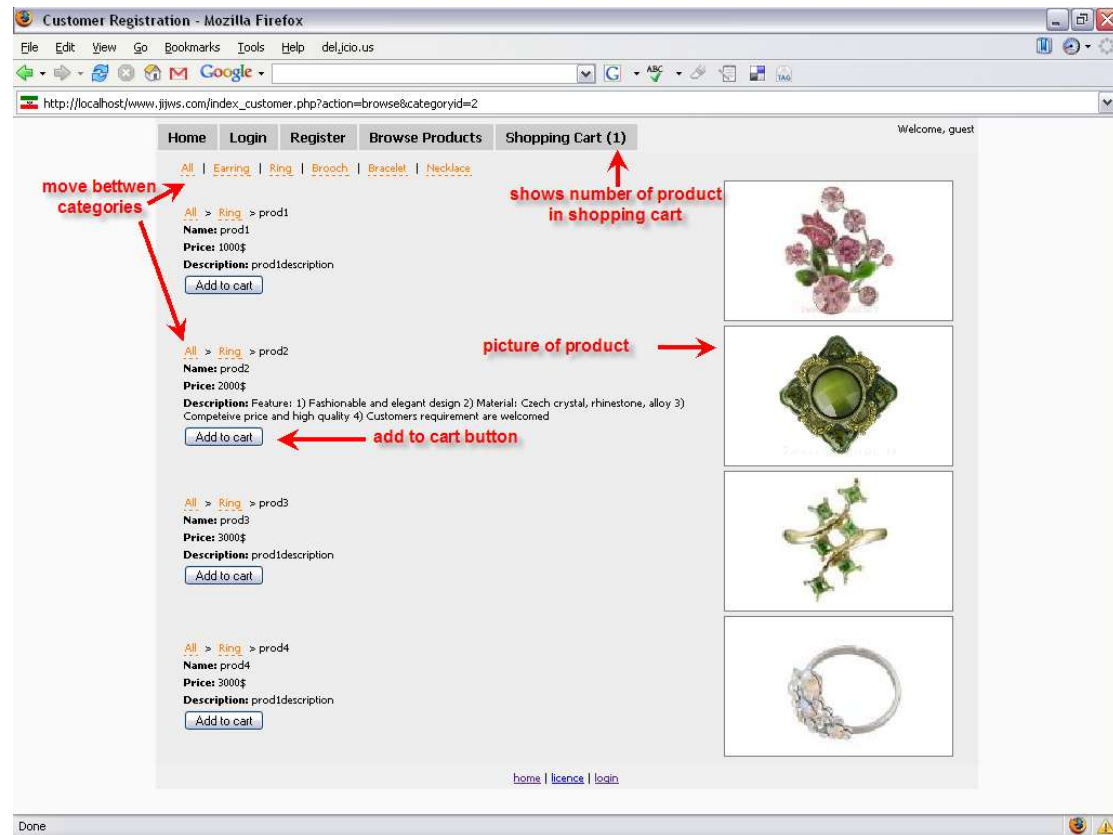


Figure 17 – Browse page

7.3.3 Shopping basket

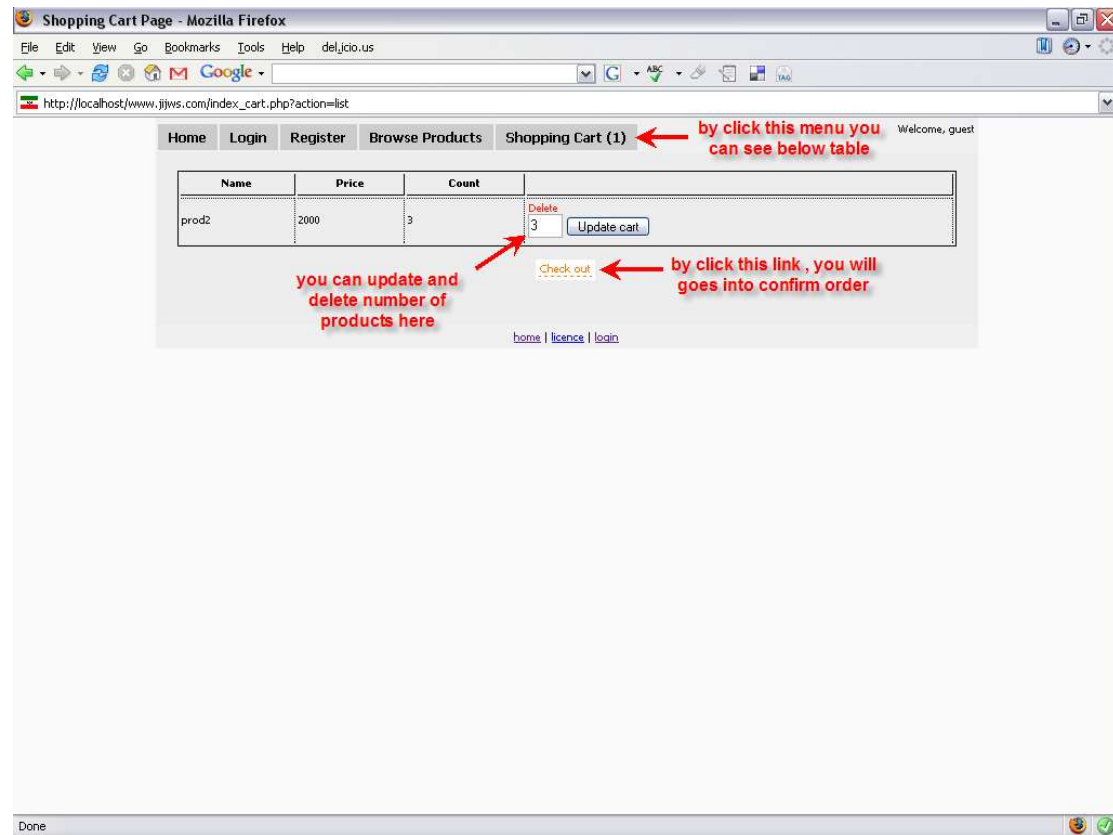


Figure 18 -Shopping Basket

7.3.4 Confirm Order

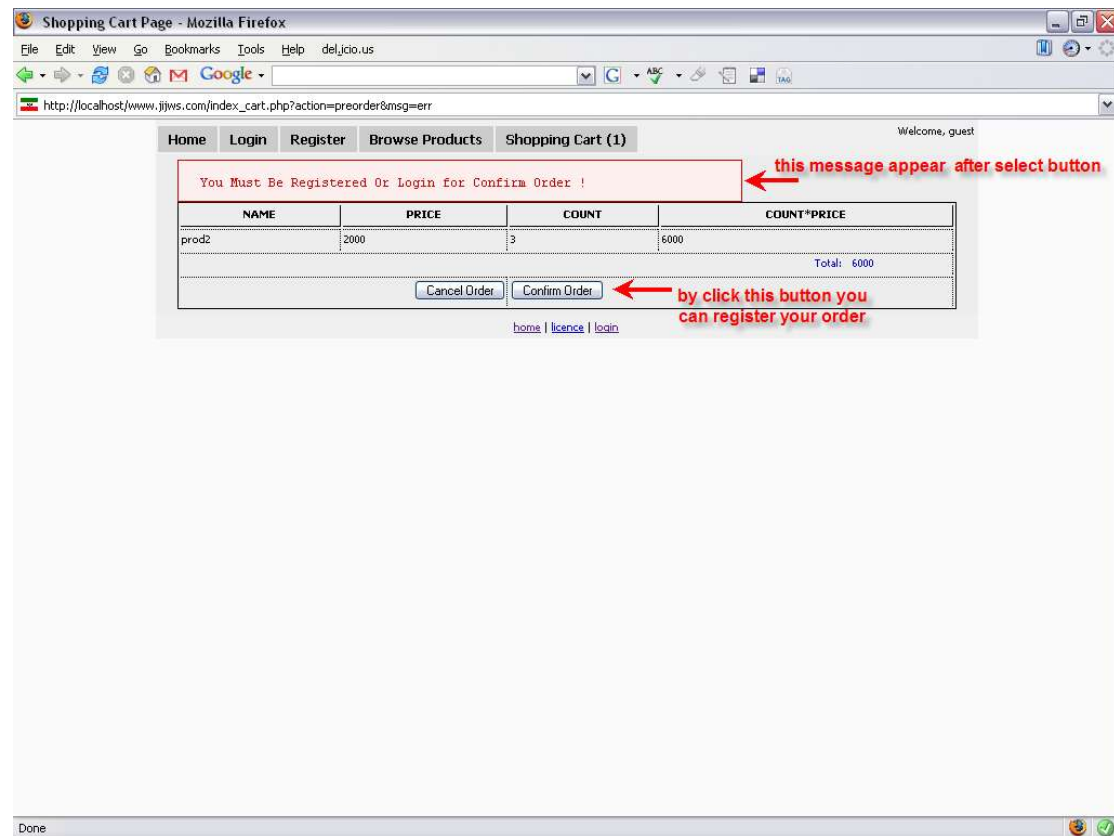


Figure 19 – Confirm Order

7.3.5 Registration Page

Customer Registration - Mozilla Firefox

File Edit View Go Bookmarks Tools Help delicio.us

http://localhost/www.jjws.com/index_customer.php

Home Login Register Browse Products Shopping Cart (1) Welcome, guest

- Username Have Invalid Chars.
- Password must be more than 4 chars
- Email is Invalid.
- Date Format Must Be Like Year-Month-Day (e.g. 1983-05-20)
- Please Select Category .

different types of validation

Customer Registration Form

Username*: saied's

Password*:

Confirm Password*:

Name_Family*: saeid moradi

Address : USA west 186 !

E-mail*: saeid3@gmail

Work Phone NO : 555189234

Cell No : 1234

Date Of Birth*: 2007-25-02

Category*:

Register

home | licence | login

Done

Figure 20 – Registration Page

7.3.6 After Registration

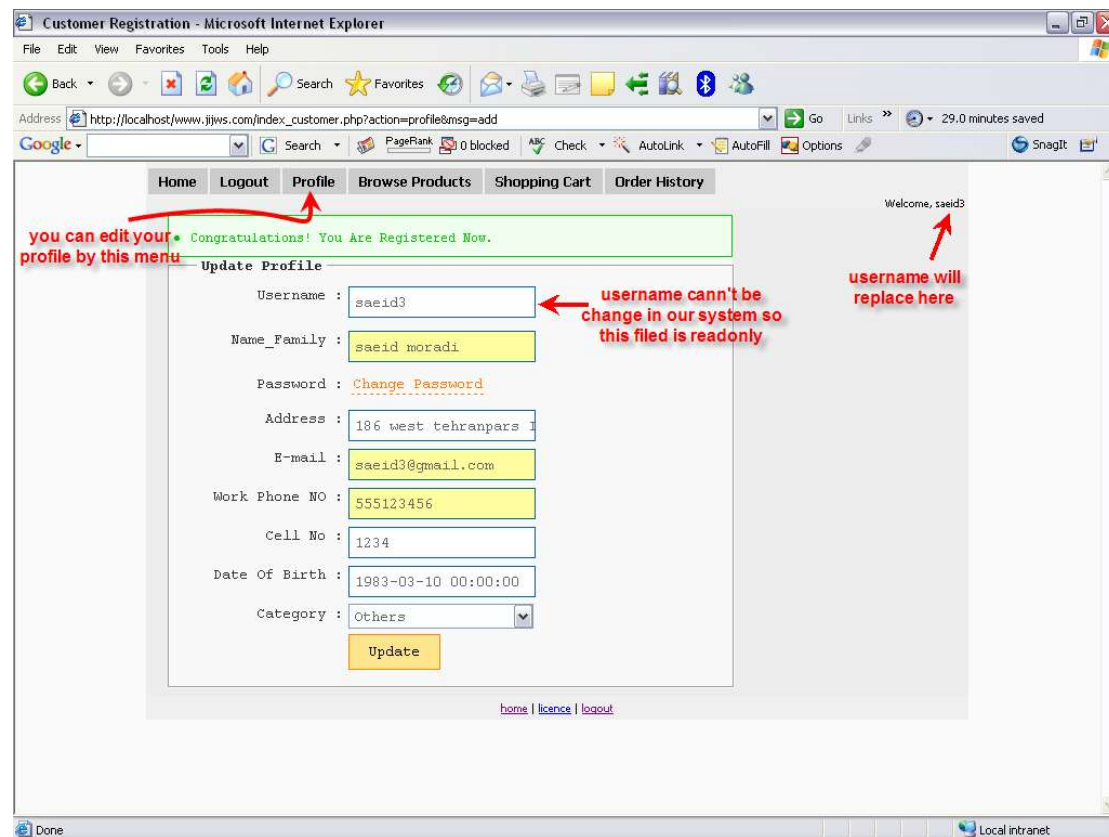


Figure 21 – After Registration

7.3.7 Order History

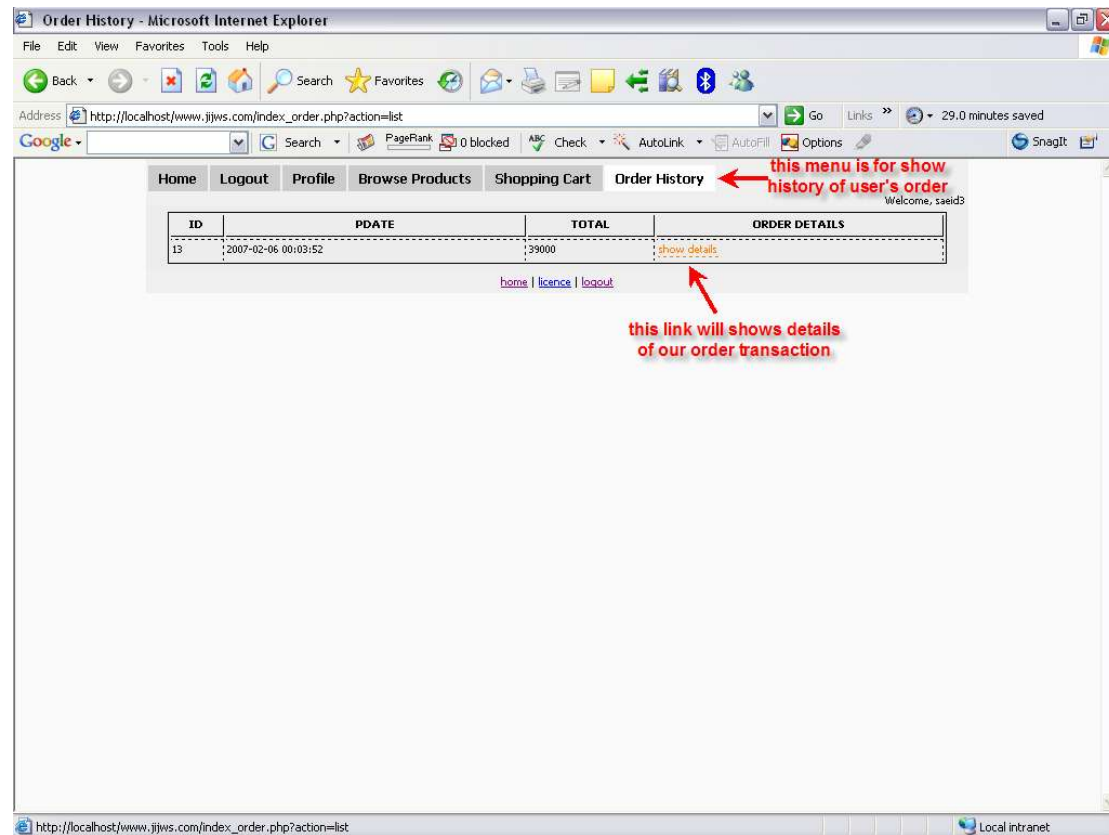


Figure 22 – Order History

7.4 Troubleshooting and FAQ

this application is compatible with different browser ?

We try to support and test between two common browser Internet Explore And Mozilla FireFox .

can we change look and face of project easily ?

Yes, easily , because UI and HTML tags have design base on css .

If We change our database by ADOdb into MSSQL , we must change some part of our code ?

No , Our code and queries are fix with this change.

Which version of Adobe Acrobat Reader is more suitable for have all features of this document ?

Adobe Reader 8.0 .

When I import project into eclipse , i see some error logs in console , is it error of project ?

No, for improve this errors you must do below work :

In each switch eclipse will recompile a source code. And in second time you must don't have any error .

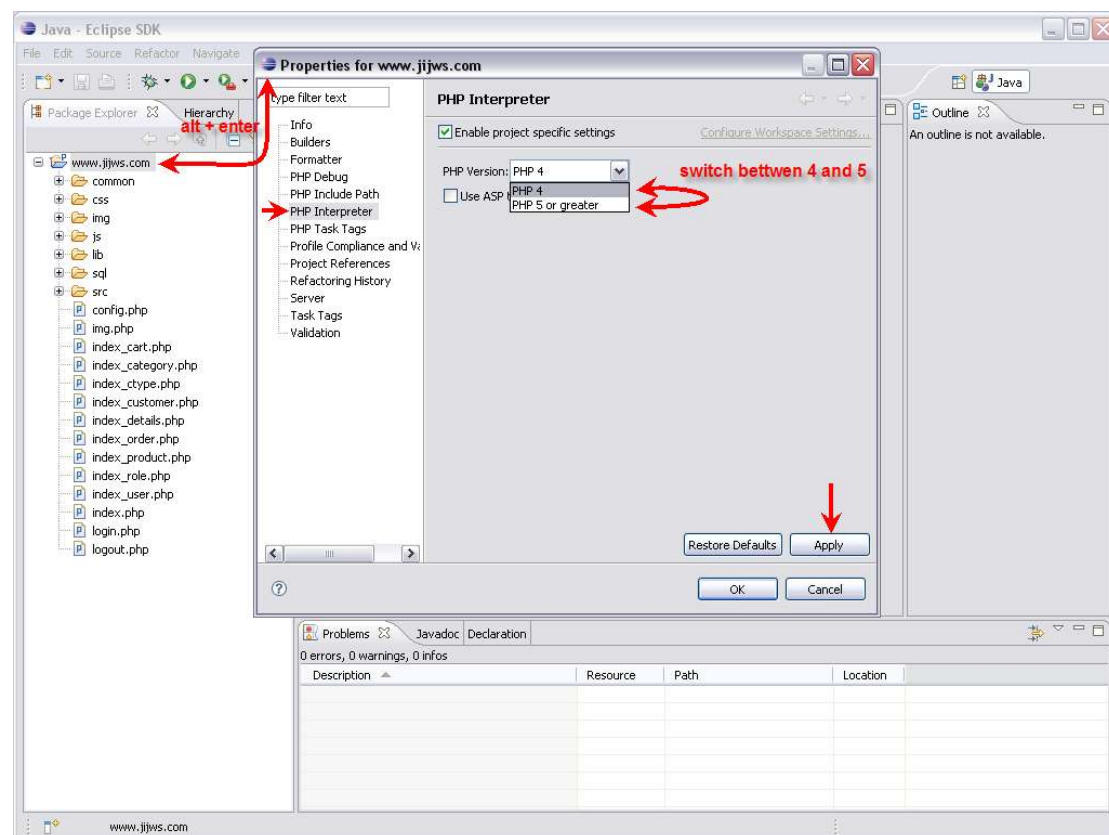


Figure 23 – Error log in project