**Chapter 1**

**Introduction**

A review website is that kind of website where you can find some reviews on various categories given by general people. Review-BD is the same. But unlike other review website Review-BD gives users to post their own review. For that it has registration system for the the users. Users will post their review which will show in the main page of the website. User can post these reviews category wish. They can read any review of other real people who posted it. There is also search option to search any kind of review they want to read and they can browse posts using the category option.

Finally we have built Admin panel where Admin can take action against violators, violence posts and violence comments. Moreover, in our admin panel Admin can see the total number of users have been assigned and total number of posts have been posted.

**1.1 Motivation:**

For our project we consider to develop this review website for two reasons.

1. There are lots of websites on different topic but there is no website that will give us a platform for all kind of reviews.
2. There are no good review websites that will provide reviews related to Bangladeshi products.
   1. **Purpose of the project:**

To change the concept of review system that exists. Giving people a great experience about anything they want to consume.

* 1. **Objective:**

A review website where people can register, post and view others review to get an idea about what they are going to consume.

* 1. **Outline:**
* Introduction
* The system
* Design Specification
* Code Specification
* Future Work
* Conclusion

**Chapter 2**

**The System**

**2.1 User Activities:**

There are two kind of user of this system.

1. User who have an account.
2. User who does not have an account.

User who have an account:

1. They can read any post including categorical post feature.
2. They can comment on any post
3. They can report a post
4. They can give a review. Using the Write Review option.
5. They have their own profile
6. They can search
7. They can share to social network like Facebook

User who don’t have an account:

1. They can read any post including categorical post feature.
2. They cannot comment on anything.
3. They can register to become a member and get all the features that are available for the user who have an account.
4. They can search.
5. They can share to social network like Facebook.

**2.2 Admin Activities:**

For admin there are different pages where

1. Admin can logging
2. Manage users posts
3. Manage users ids
4. Manage user comments
5. See total post and users and so on.

**Chapter 3**

**Requirement Analysis**

The website required the following requirement. There are mainly two actors such as user and admin.

**3.1: Data Requirement**

During the requirement analysis the following data have been identified as data requirement.

* The users can read any post on the site and share it to the Facebook.
* If the users want to give their own post or comment to any post they need to login or create a new account.
* Logged in users can post and comment. They will have their own profile and they can upload their images.
* Logged in users can report any post.
* When users are giving a post, he must select a category and give some tag on this. He may choose some subcategory to make the post more specific.
* Any visitor can see the notice.
* Admin will login to the admin panel to manage the website.
* He can see how many post or users are there.
* He can delete any user.
* He can delete any post.
* He can delete any comment.
* Admin can give any notice regarding the website.

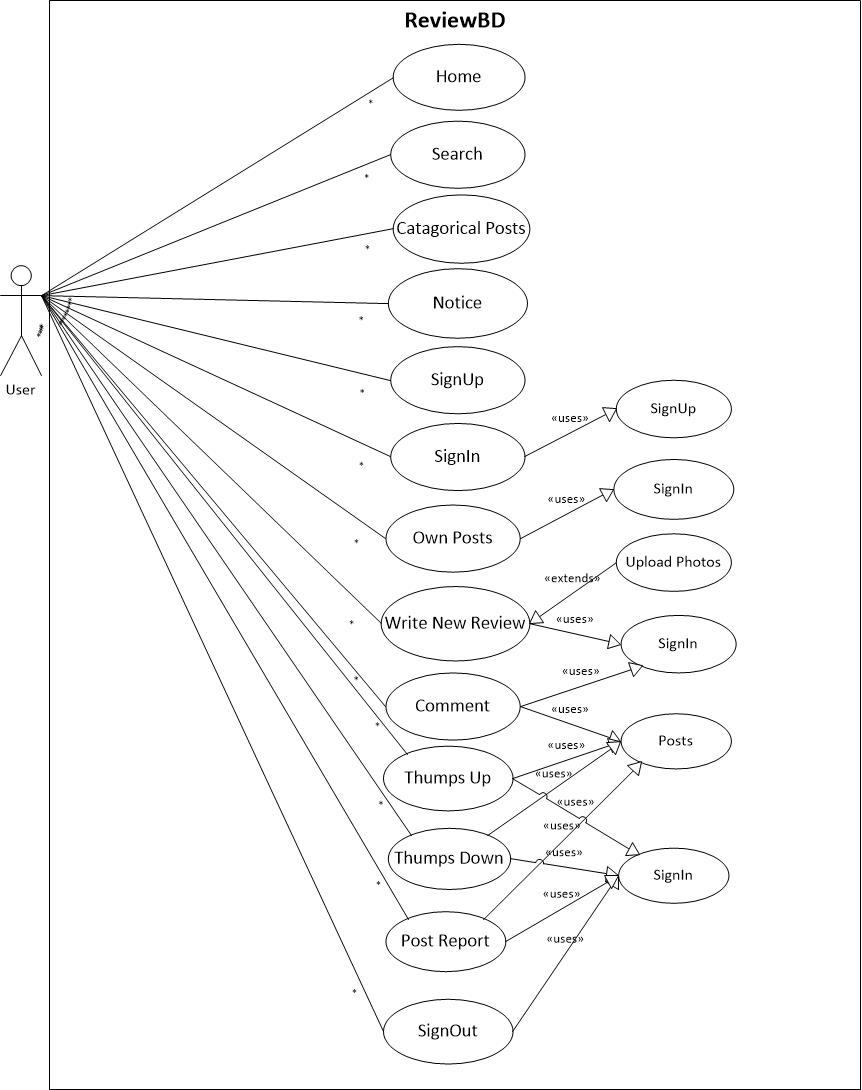
**3.2: Physical Design**

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified or authenticated, how it is processed, and how it is displayed as In Physical design; the following requirements about the system are decided.

* Input requirements
* Output requirements
* Storage requirement
* Processing Requirements

**3.3 Use case Diagrams**

**3.3.1 Use Case Diagram for User:**

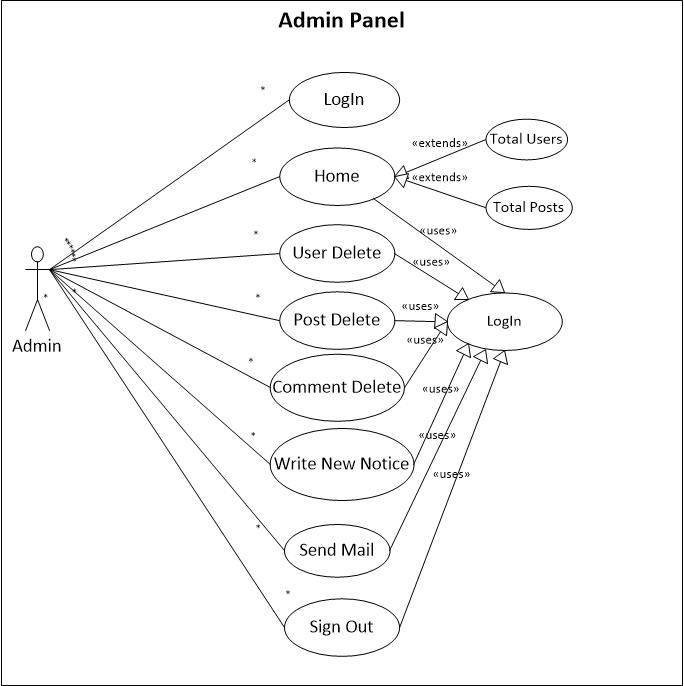
****

**Fig 3.1: Use Case Diagram for User Management.**

**Description:** This is the Use Case Diagram of a user. This diagram shows the interaction between User and the site.

* **Home:** Users can go to home page and can see all contents of the homepage.
* **Search:** Using search option user can find any of his/her desired post.
* **Categories:** By using categorical post Users can find post by category. User doesn’t need to go through all posts.
* **Notice:** Users can see the notice which would have posted by admin.
* **Sign up:** To become a registered member Users have to be registered.
* **Sign-in:** To write a new review or put a comment on someone posts, user has to be logged in. Here is a Uses use case included where base case is depends on the Uses use case. Without Uses use case the base can’t go through the sign in.
* **Own Posts:** So far you have been posted review on our site you can see those posts but you have to be registered member of this site. Therefore, Uses use case is included here.
* **Write new review:** Users can write a new review for the site. Here is included another use case name is Extend use case which is depend on the base case. It is opposite to Uses use case.
* **Comment:** Users can put comment on the posts but User has to be logged in.
* **Sign-out:** If you are not logged in how can you sign out from the system. Therefore base case is depends on the Uses use case.

**3.3.2 Use Case Diagram for Admin:**

****

**Fig 3.2: Use Case Diagram for Admin Management.**

**Description:** This is the Use Case Diagram of an Admin. This diagram shows the interaction between Admin and the Admin Panel.

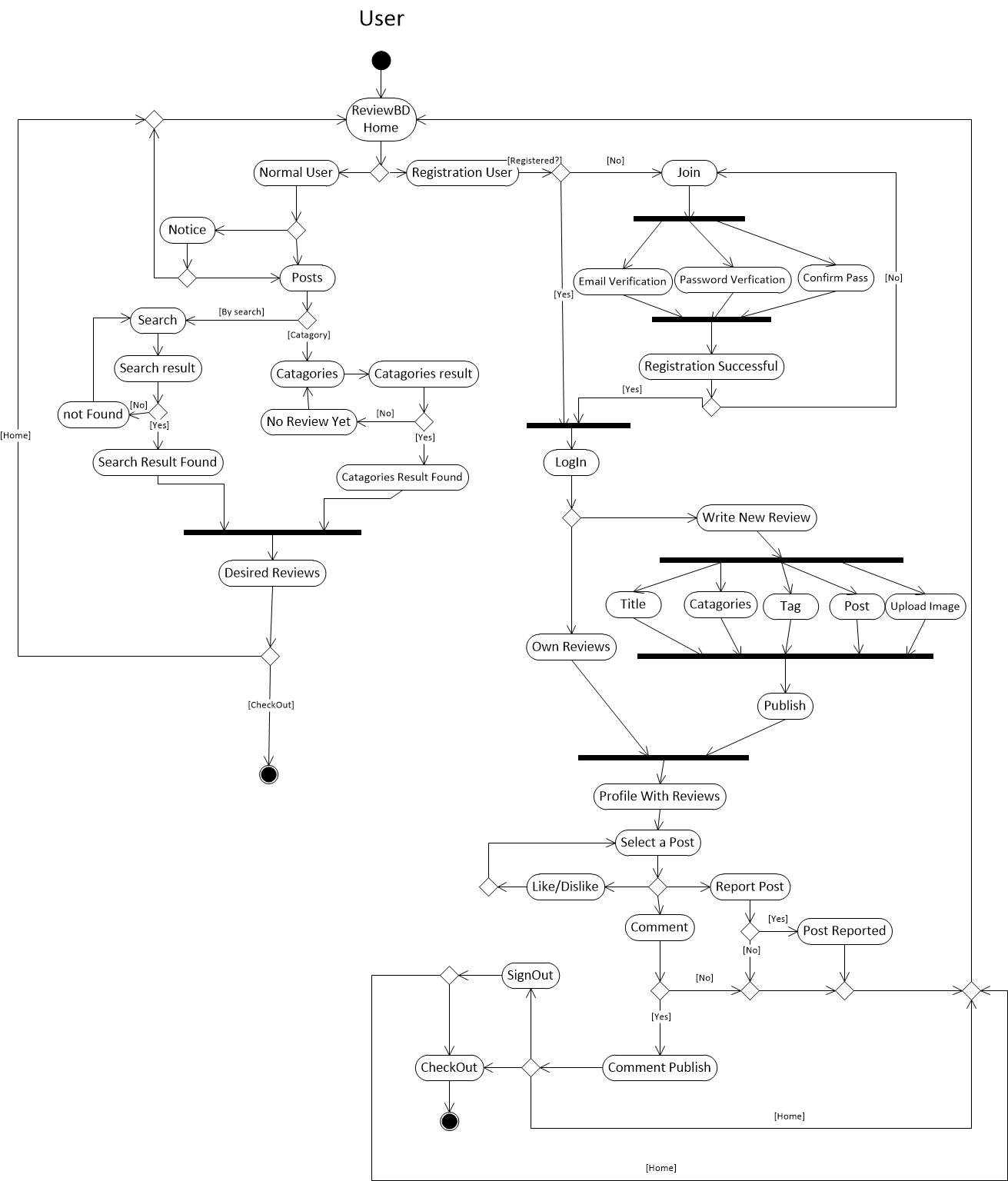
* **Login:** Admin can log in to the Admin panel.
* **Home:** Admin can go through the home page contents. Here are some extending use cases whose are completely depend on home page. Moreover, there is included a Uses use case as well.
* **Users Delete:** Admin can take action against violators.
* **Posts Delete:** Admin can delete post which posts are irrelevant according to our site.
* **Comments Delete:** Admin can take action against violence comments.
* **Sign out:** Final step is sign out option for the Admin.

**3.4 Activity Diagrams**

Let's start by describing the basic notation that I've used in Figures 3.3 and 3.4:

* **Initial node**. The filled in circle is the starting point of the diagram. An initial node isn't required although it does make it significantly easier to read the diagram.
* **Activity final node**. The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes.
* **Activity**. The rounded rectangles represent activities that occur.
* **Flow/edge**. The arrows on the diagram. Although there is a subtle difference between flows and edges I have never seen a practical purpose for the difference although I have no doubt one exists. I'll use the term flow.
* **Fork**. A black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity.
* **Join**. A black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.
* **Decision**. A diamond with one flow entering and several leaving. The flows leaving include conditions although some modelers will not indicate the conditions if it is obvious.
* **Merge**. A diamond with several flows entering and one leaving. The implication is that one or more incoming flows must reach this point until processing continues, based on any guards on the outgoing flow.

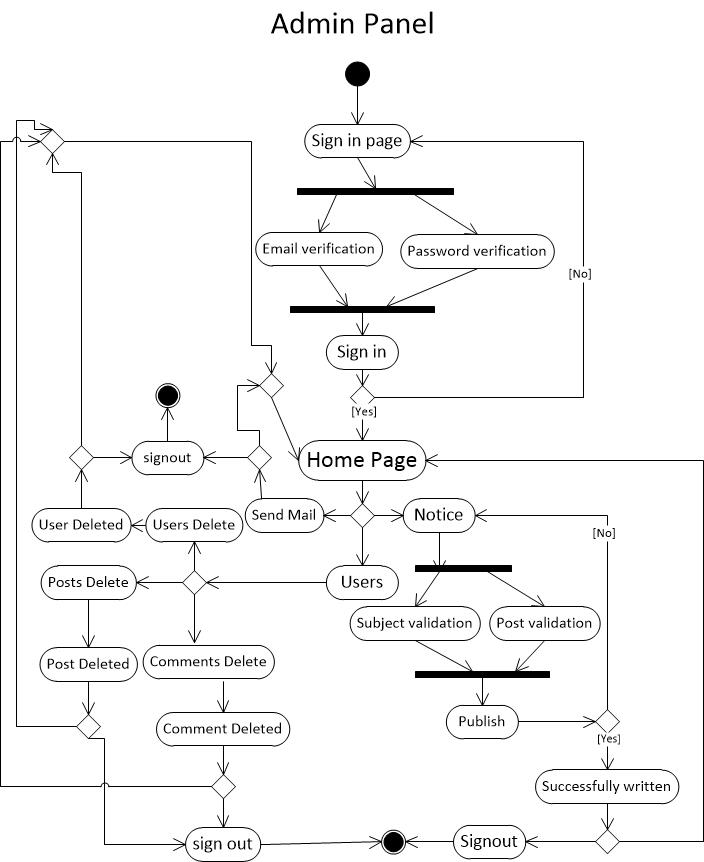
**3.4.1 Activity Diagram for User.**

****

**Fig 3.3: Activity Diagram for User Activities.**

**Description:** This is the Activity Diagram of a User. Here we have showed some activities about normal users and logged in users. The only difference between them, logged in user can go through in all sections as well as a normal user whatever does but normal users have some restrictions to go through in all sections.

**3.4.2 Activity Diagram for Admin:**

****

**Fig 3.4: Activity Diagram for Admin Activities.**

**Description:** This Activity diagram shows how the admin interact with different activities and how they jump one step to another step additionally come back to home page and sign out from the system.

**3.5 Software Requirement:**

To implement this web application any technologies that are used are open sources. We are discussing about them below.

**3.5.1 HTML:**

Html is a markup language for describing the web documents. In our website we use HTML5. Every web page you see on the Internet, including this one contains HTML code that helps format and show text and images in an easy to read format. Without HTML a browser would not know how to format a page and would only display plain text with no formatting that contained no links. [1]

**3.5.2 CSS:**

We use CSS3 to give our HTML a shape. CSS stands for cascading style sheet. [2]

**3.5.3 BOOTSTRAP [Version: 2]:**

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes. [3]

**3.5.4 JavaScript:**

JavaScript is the programming language of HTML and the Web Programming makes computers do what you want them to do. We use JavaScript for our dropdown menu using bootstrap. [4]

**3.5.5 JQuery:**

We used jQuery for our subcategories dropdown form. It is a framework of core JavaScript. [5]

**3.5.6 PHP [Version: 5.4.16]:**

PHP is probably the most popular scripting language on the web. It is used to enhance web pages. With PHP, you can do things like create username and password login pages, check details from a form, create forums, picture galleries, surveys, and a whole lot more. If you've come across a web page that ends in PHP,

Then the author has written some programming code to liven up the plain, old

HTML. PHP is known as a server-sided language. That's because the PHP doesn't get executed on your computer. But on the computer you requested the page from. The results are then handed over to you, and displayed in your browser. [6]

**3.5.7 PHP CodeIgniter Framework [Version: 2.2.2]:**

CodeIgniter is an Application Development Framework - a toolkit - for people who build web sites using PHP. Its goal is to enable to develop projects much faster than you could if you were writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. Code Igniter lets you creatively focus on your project by minimizing the amount of code needed for a given task. [7]

**3.5.8 Apache Server (XAMPP) [Version: 1.8.2]:**

The Apache HTTP Server, commonly referred to as Apache is a [web server](http://en.wikipedia.org/wiki/Web_server) application notable for playing a key role in the initial growth of the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web). Apache is developed and maintained by an open community of developers under the auspices of the [Apache Software Foundation](http://en.wikipedia.org/wiki/Apache_Software_Foundation). Most commonly used on a [Unix-like](http://en.wikipedia.org/wiki/Unix-like) system, the software is available for a wide variety of [operating systems](http://en.wikipedia.org/wiki/Operating_system), including [Unix](http://en.wikipedia.org/wiki/Unix), [FreeBSD](http://en.wikipedia.org/wiki/FreeBSD), [Linux](http://en.wikipedia.org/wiki/Linux), [Solaris](http://en.wikipedia.org/wiki/Solaris_(operating_system)), [Novell NetWare](http://en.wikipedia.org/wiki/Novell_NetWare), [OS X](http://en.wikipedia.org/wiki/OS_X), [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), [OS/2](http://en.wikipedia.org/wiki/OS/2), [TPF](http://en.wikipedia.org/wiki/Transaction_Processing_Facility), [OpenVMS](http://en.wikipedia.org/wiki/OpenVMS) and [eComStation](http://en.wikipedia.org/wiki/EComStation). Released under the [Apache License](http://en.wikipedia.org/wiki/Apache_License), Apache is [open-source software](http://en.wikipedia.org/wiki/Open-source_software).

**3.5.9 Text-Editor (Sublime Text) [Version: 2.0.2]:**

Sublime Text is the web development tool that lets you efficiently design, develop and maintain standards-based websites and applications. Sublime text provides a powerful combination of visual layout tools, application development features, and code editing support.

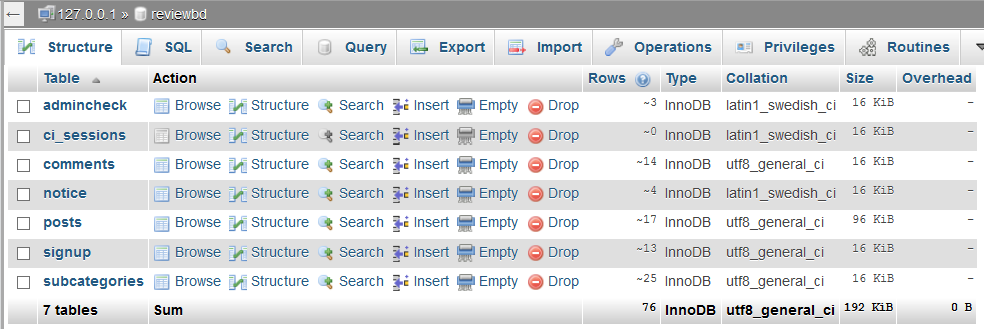
**3.5.10 MySQL [Version: 5.6.16]:**

**MySQL** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). In July 2013, it was the world's second mostwidely used RDBMS, and the most widely used open-source [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) RDBMS. It is named after co-founder [Michael Widenius](https://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, my. The [SQL](https://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query Language](https://en.wikipedia.org/wiki/Structured_Query_Language). The MySQL development project has made its [source code](https://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) agreements. We used MySQL for our database management system. [8]

**3.6 Database Design:**

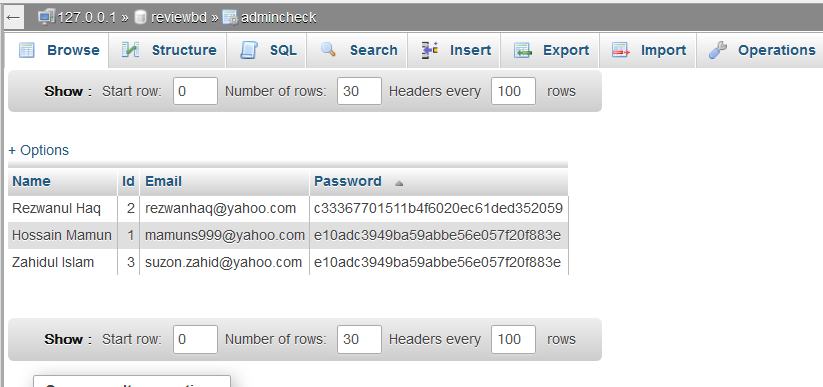
We have total 6 database tables. They are given below.

**3.6.1 List of all tables:**



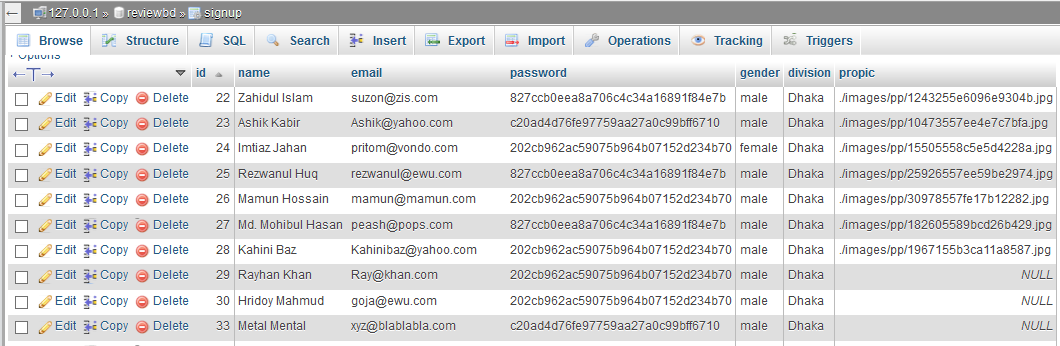
**Fig 3.5: All tables of reviewBD database.**

**3.6.2 Admin table:**



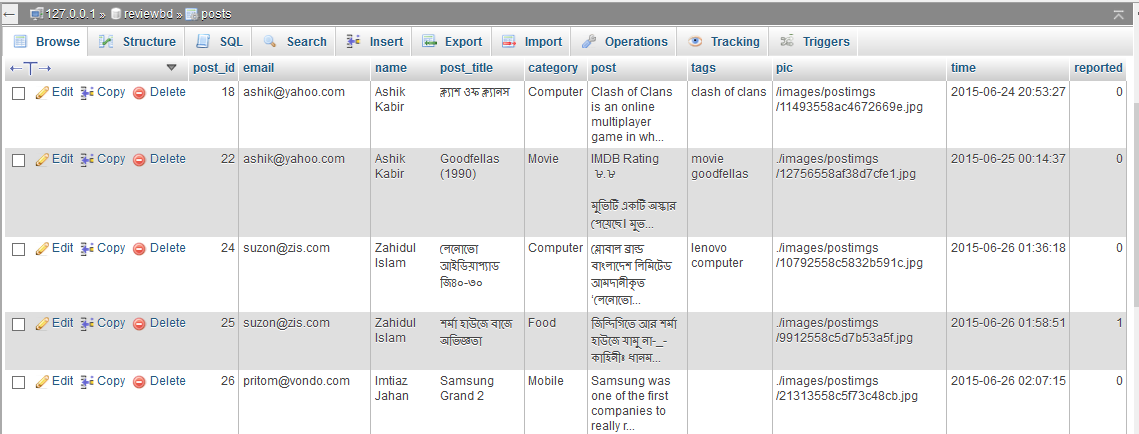
**Fig 3.6: This is admin table containing admin information.**

**3.6.3 Sign Up Table:**

****

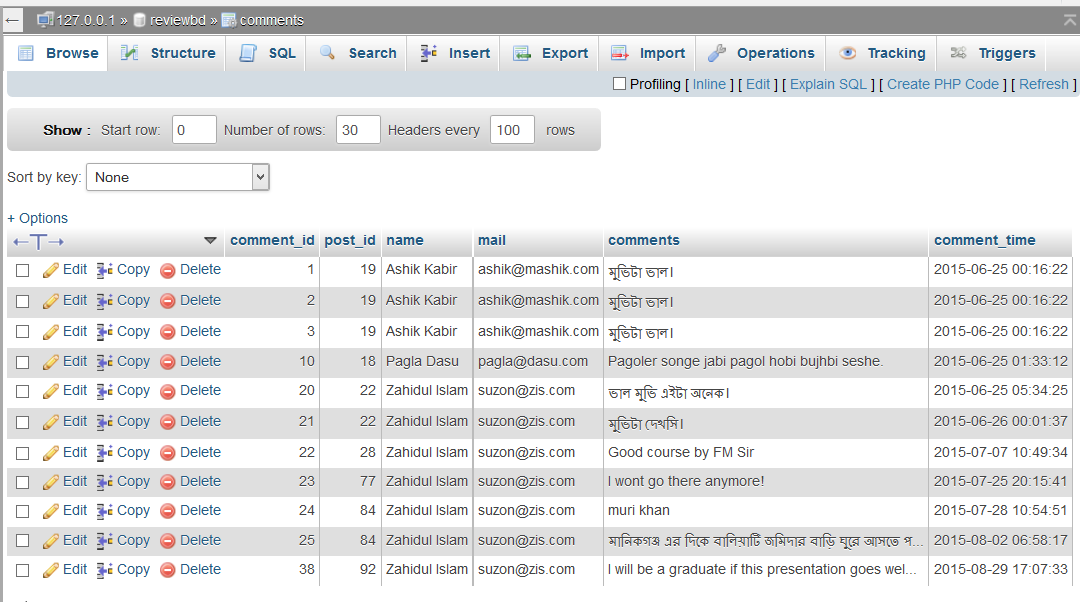
**Fig 3.7: This table contains all information of users.**

**3.6.4 Posts Table**



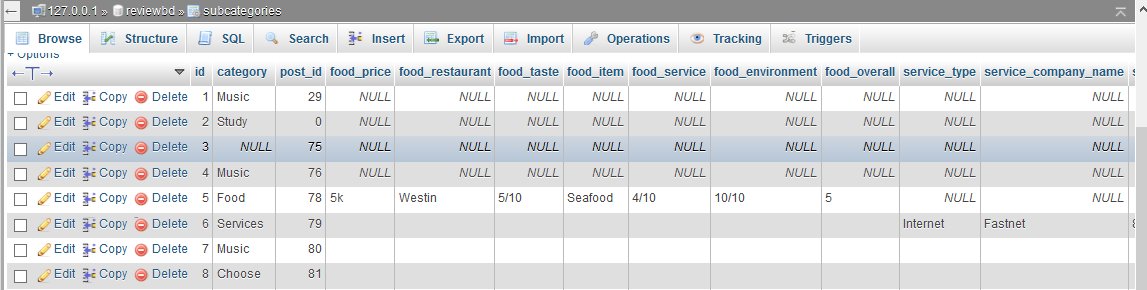
**Fig 3.8: This table contains all posts of the website.**

**3.6.5 Comments Table:**

****

**Fig 3.9: This table contains all comments using POST\_ID.**

**3.6.6 Subcategories:**

****

**Fig 3.10: This table contains all subcategories information.**

**Chapter 4**

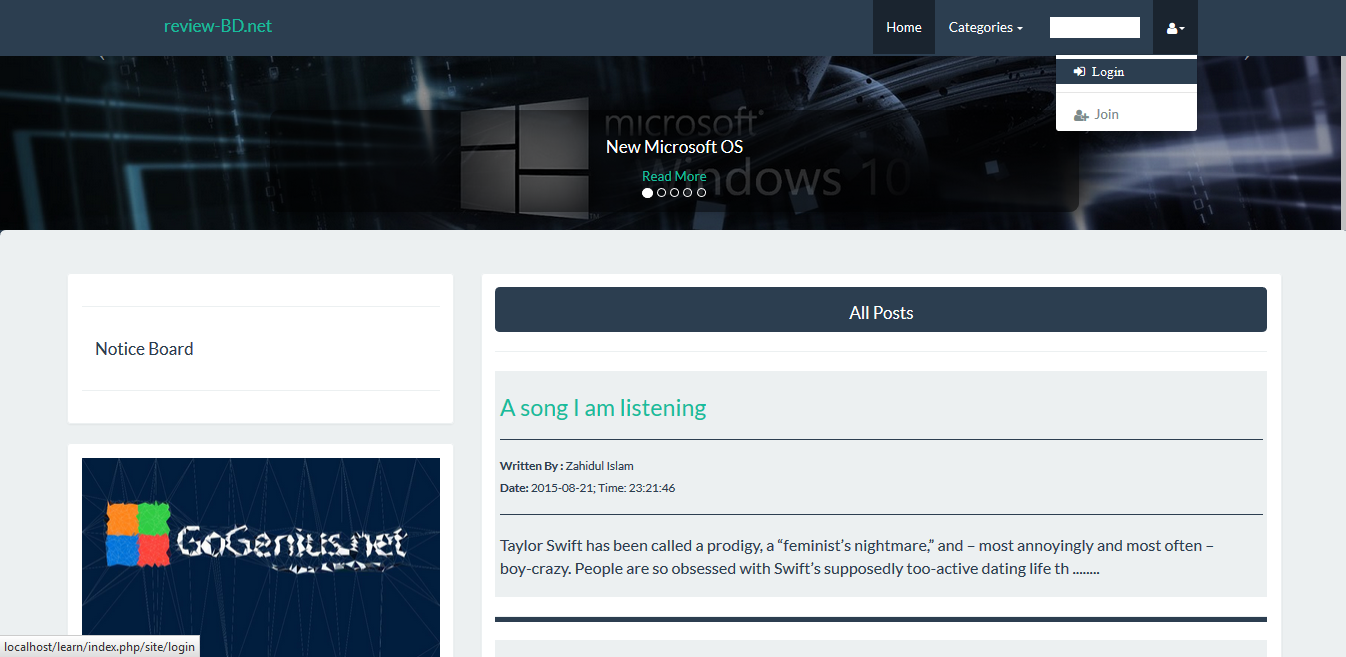
**Design Specification**

**4.1 Interface:**

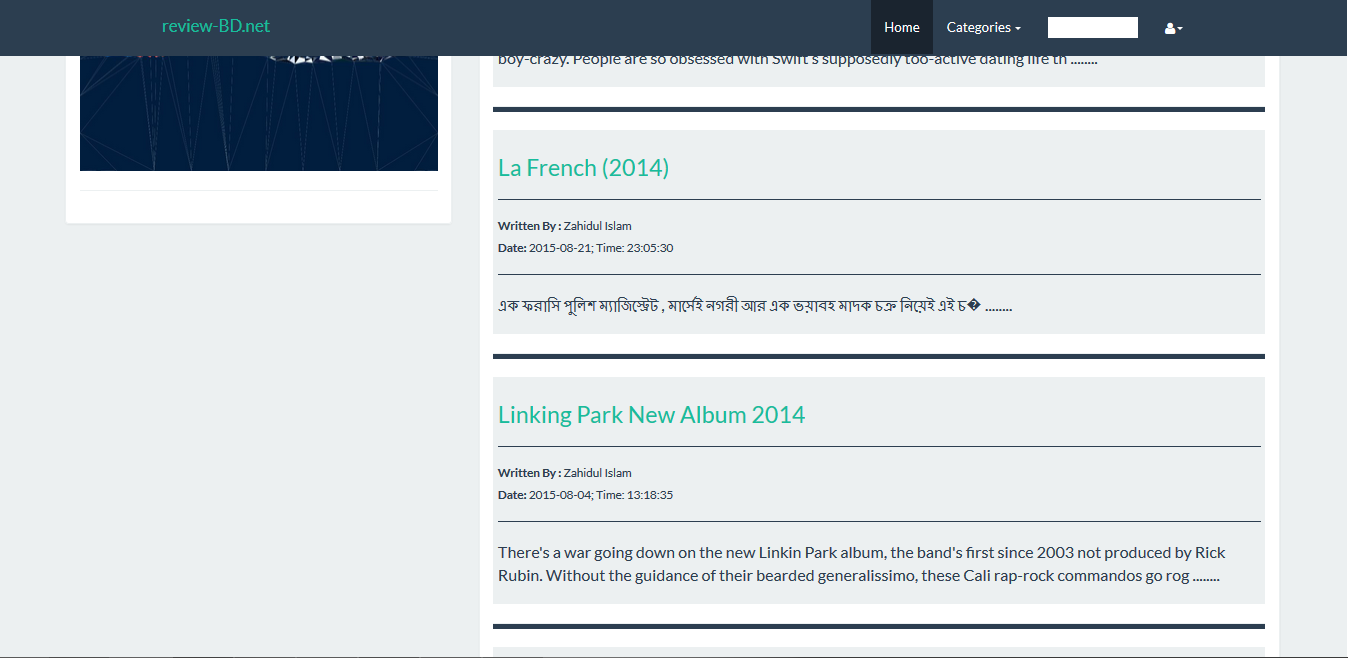
In information technology, the user interface (UI) is everything designed into an information device with which a human being may interact. We design our website user friendly. This website is fully responsive for mobile users.

**4.2 User Interface:**

**4.2.1 Home Page:**

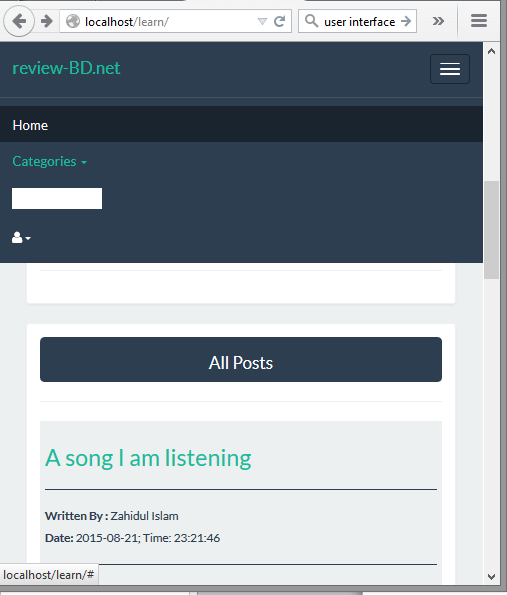
****

**Fig 4.1: Home Page**

****

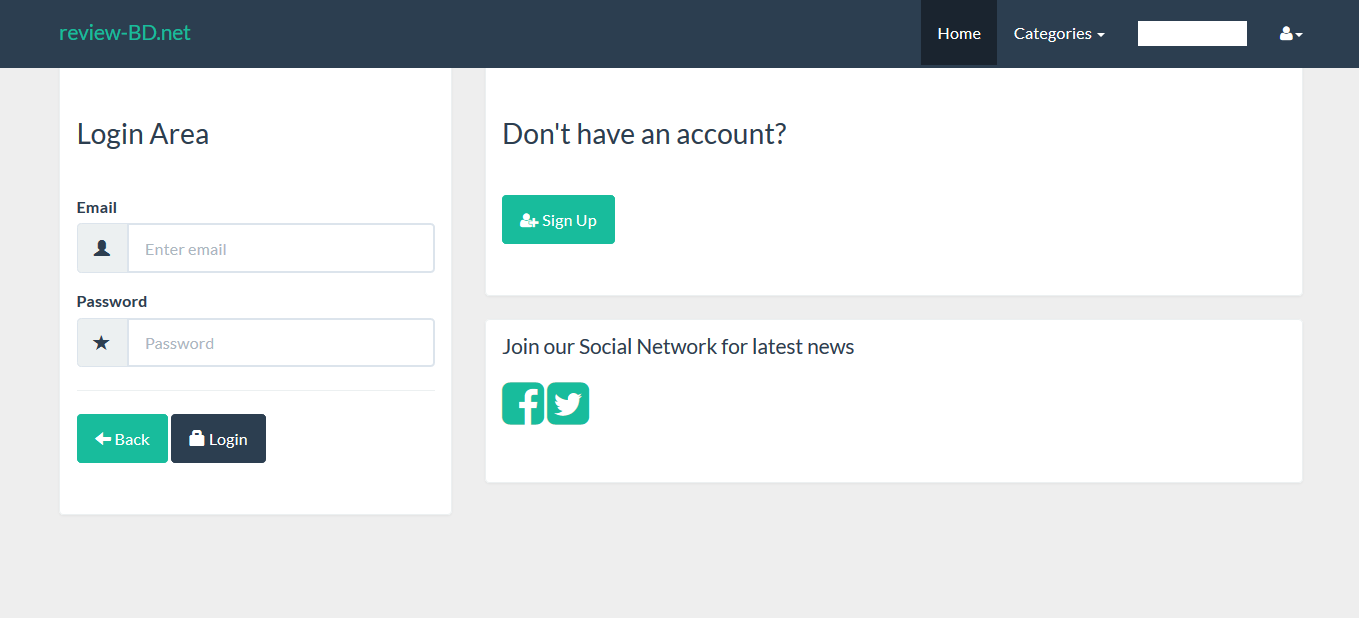
**Fig 4.1: Home Page**

**4.2.2 Mobile View:**

****

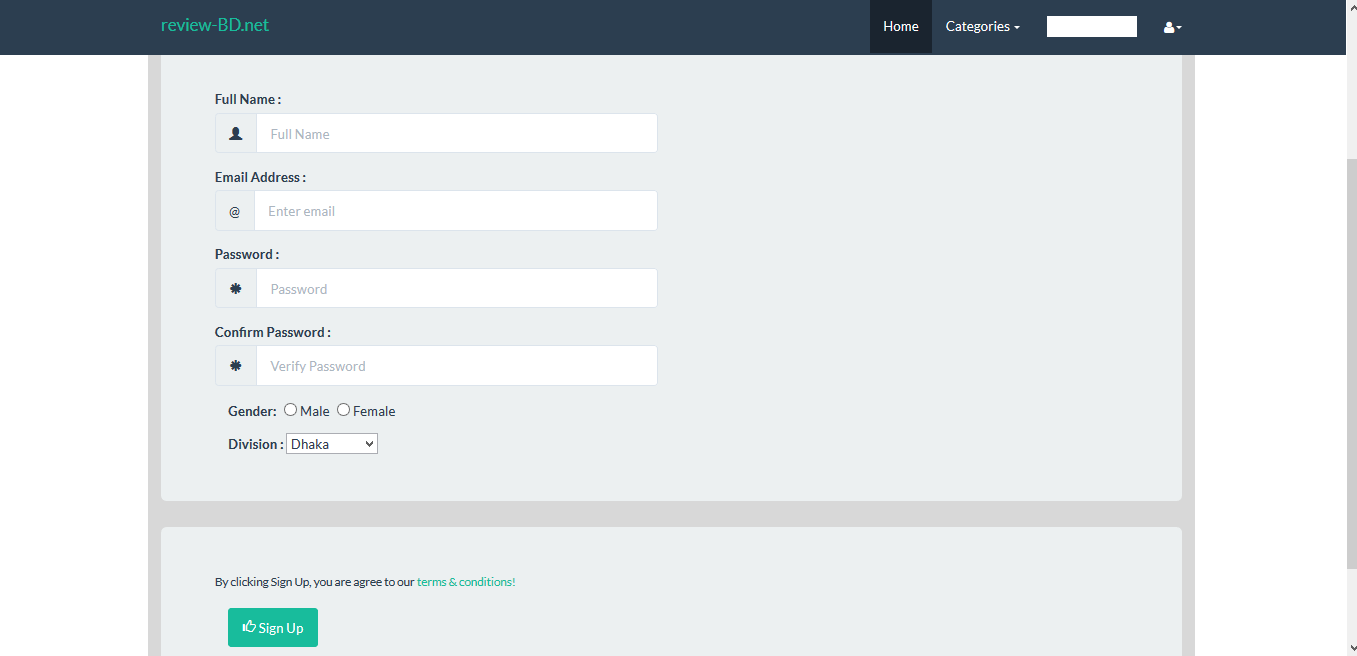
**Fig 4.2: Mobile View**

**4.2.3 Login Page:**

****

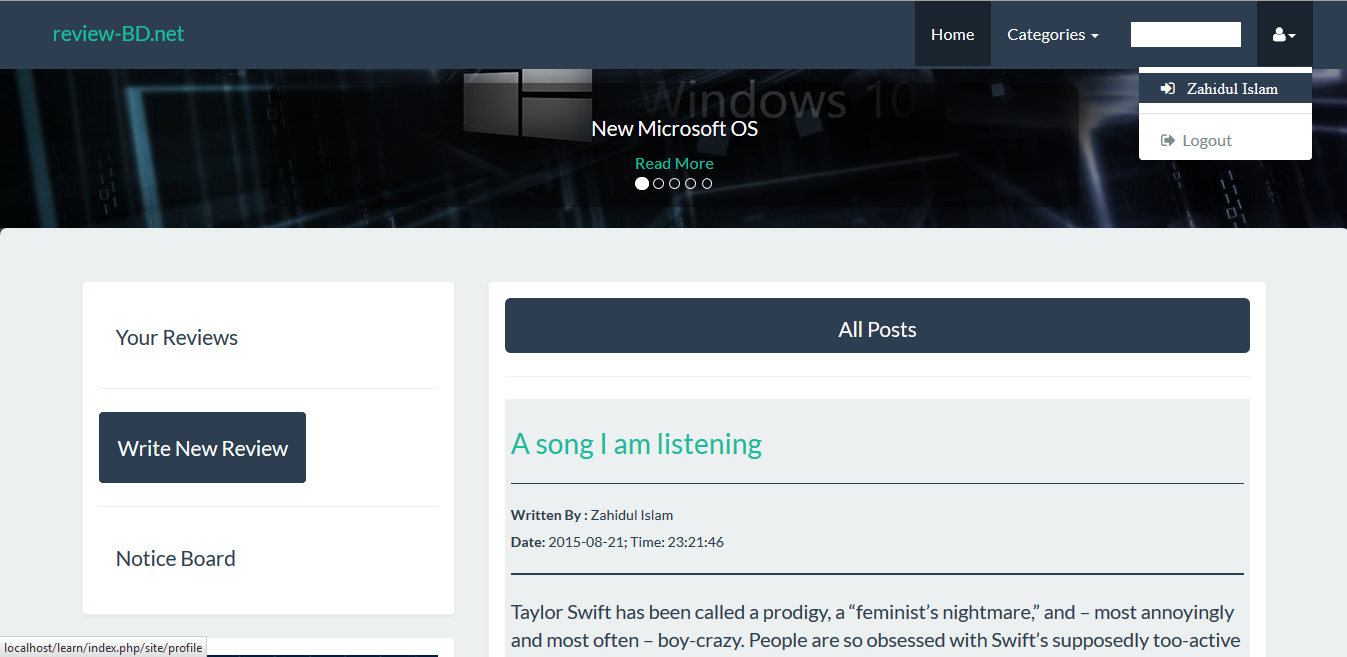
**Fig 4.3: Login Page**

**4.2.4 Signup Page:**

****

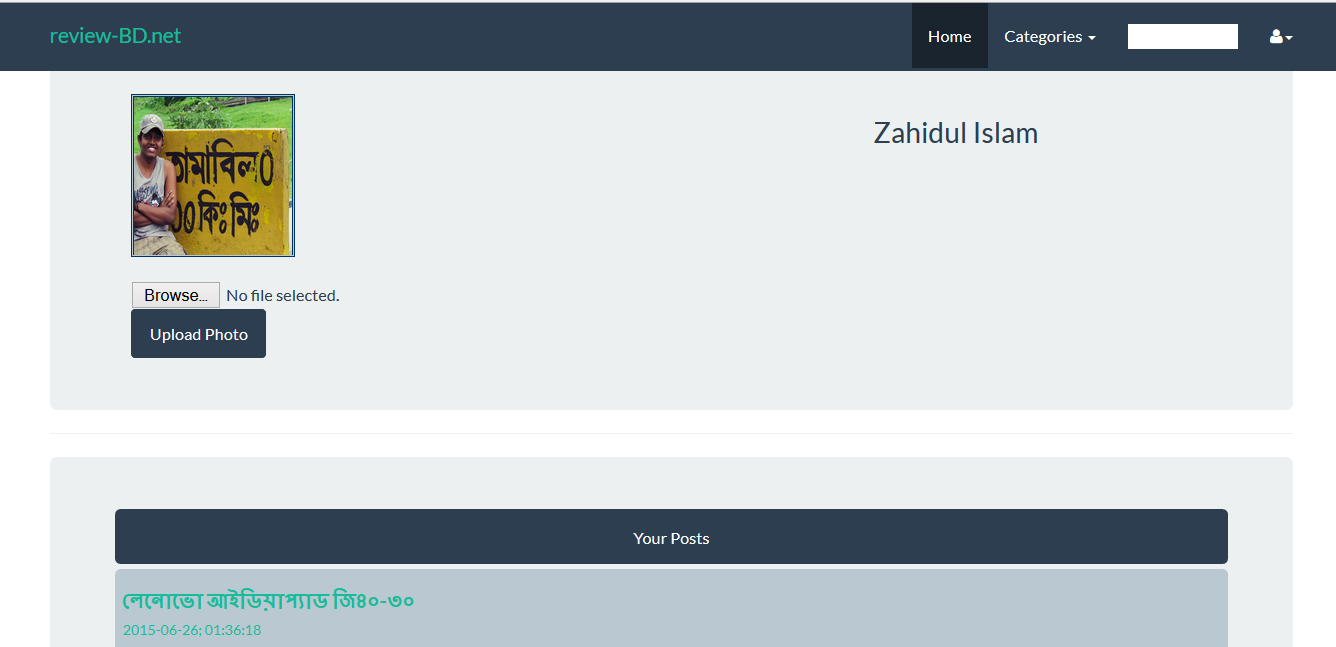
**Fig 4.4: Signup Page**

**4.2.5 After Login:**

****

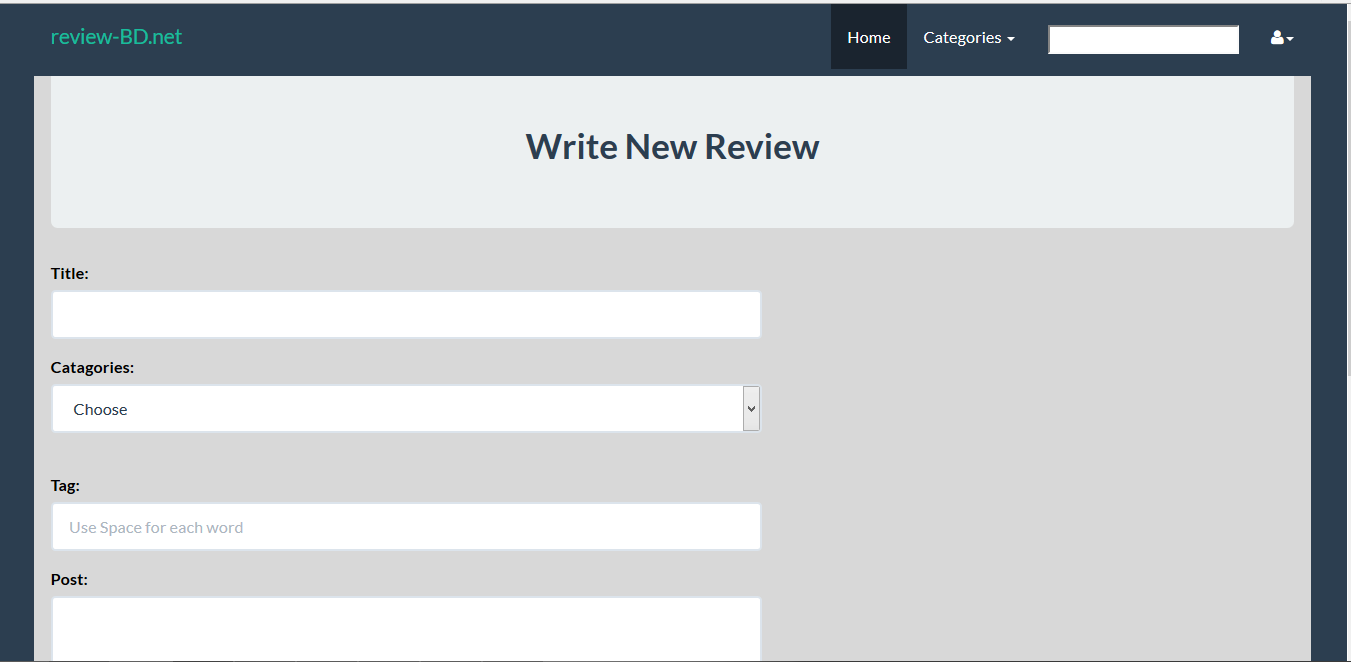
**Fig 4.5: After Login**

**4.2.6 Profile:**

****

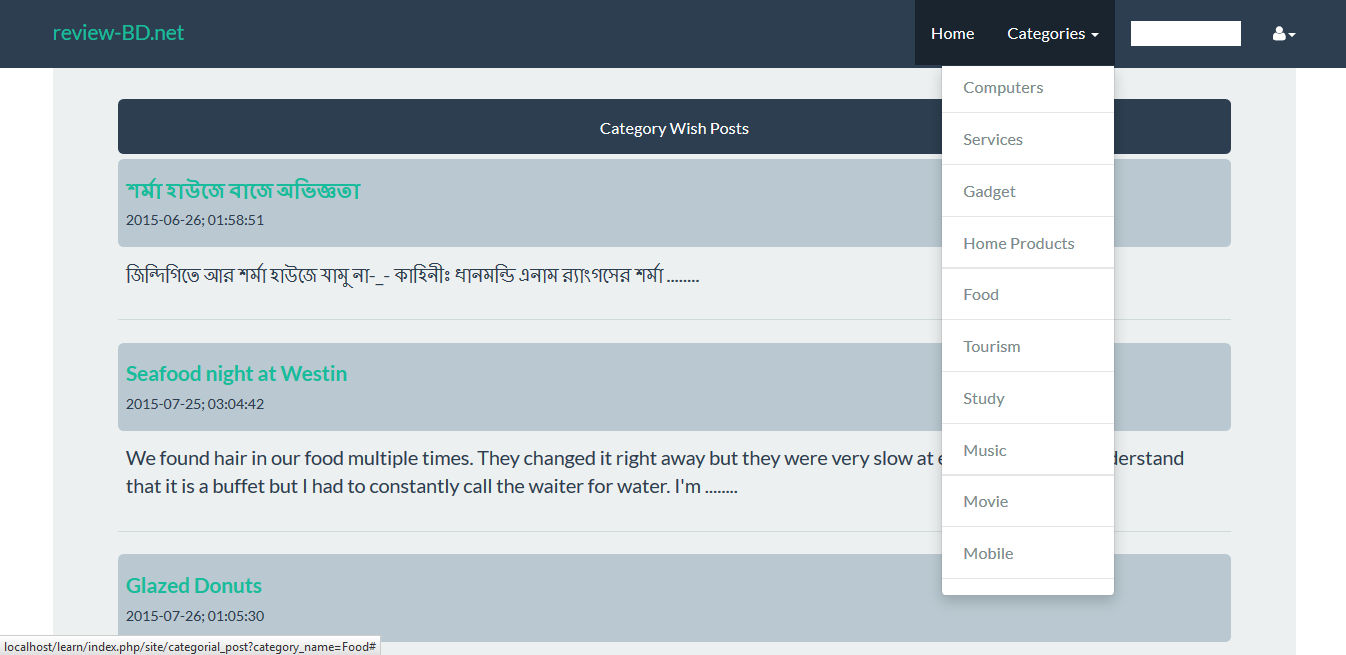
**Fig 4.6: Profile**

**4.2.7 New Review:**

****

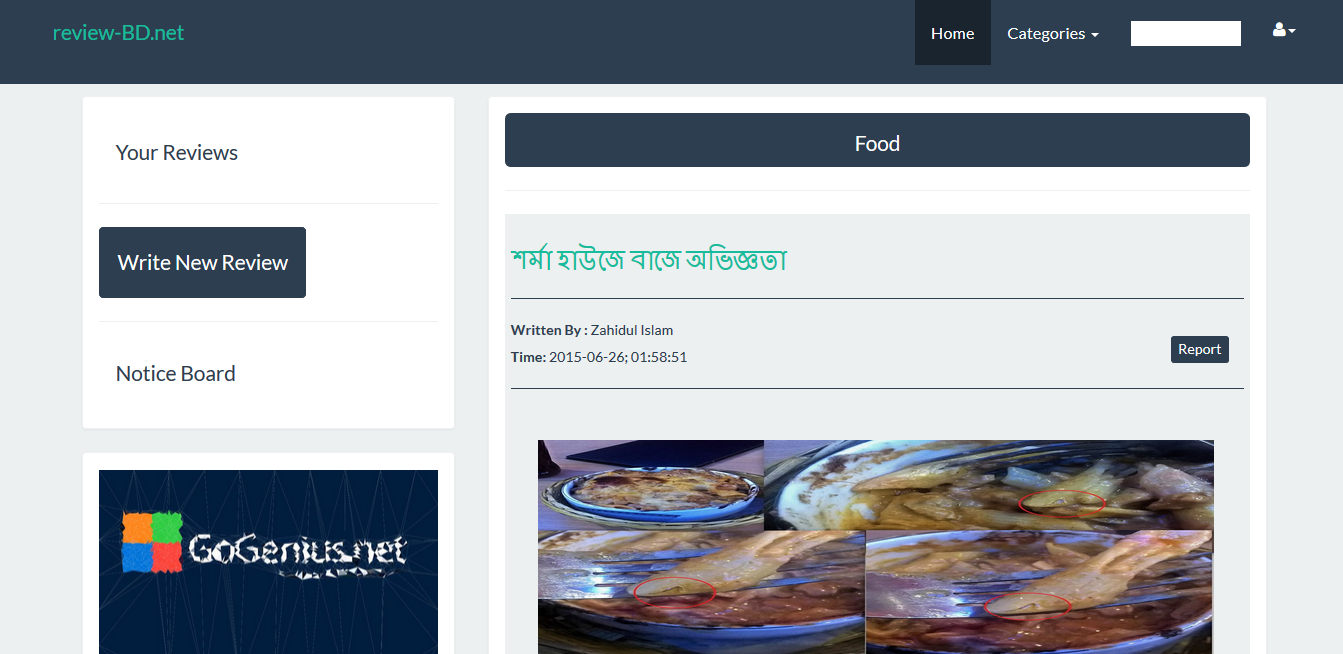
**Fig 4.7: New Review**

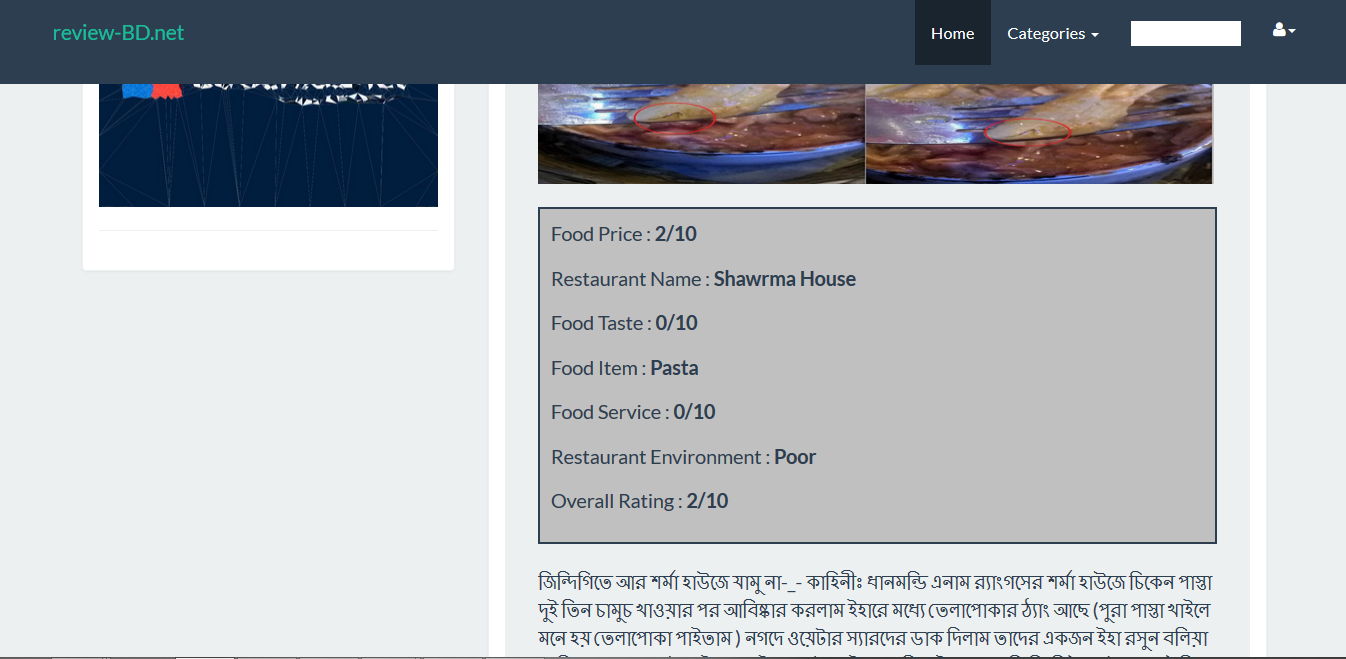
**4.2.8 Category Wise Post:**

****

**Fig 4.8: Category Wise Post**

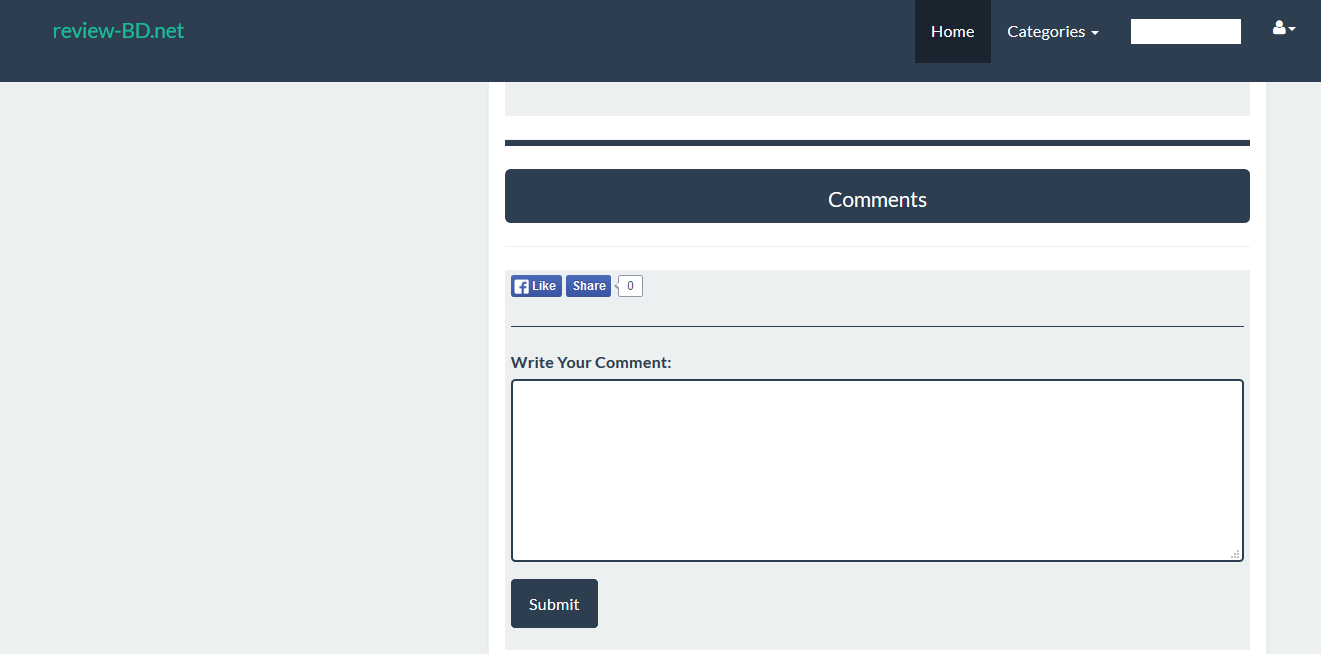
**4.2.9 Full Post:**

****

****

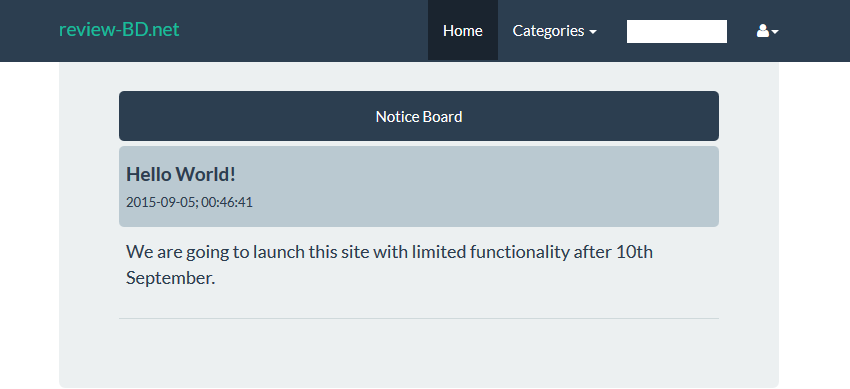
**Fig 4.9: Full Post**

**4.2.10 Comment:**

****

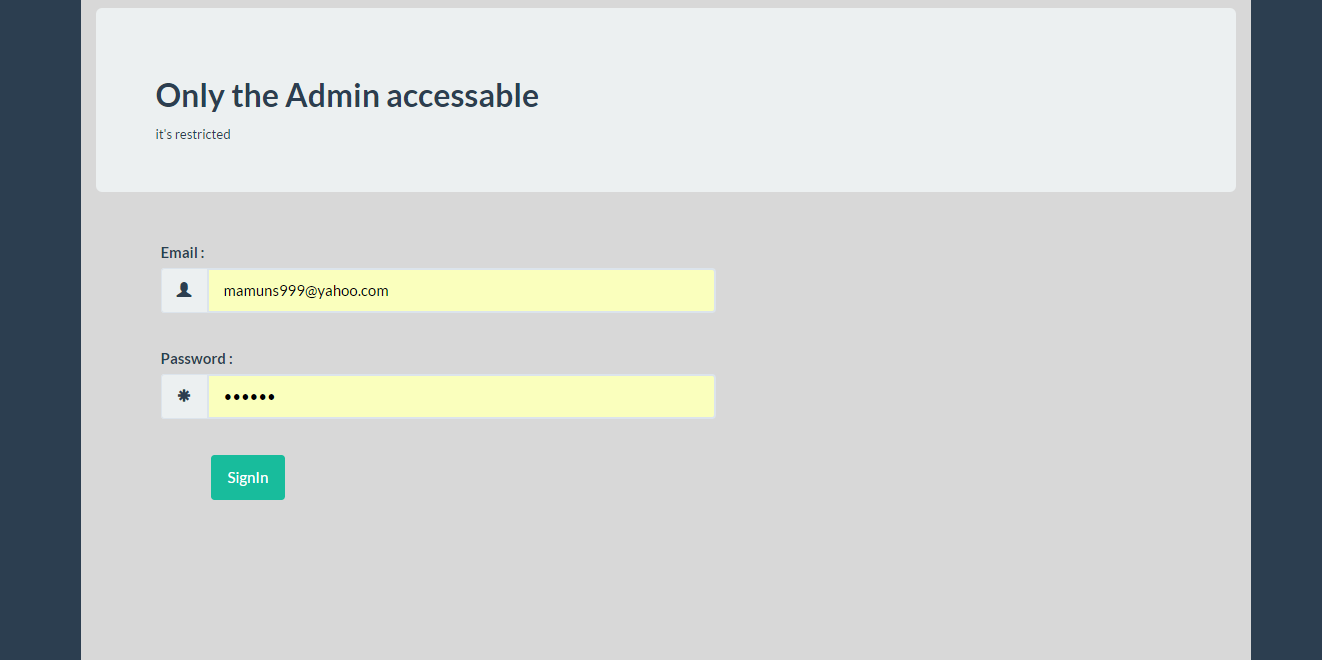
**Fig 4.10: Comment**

**4.2.11 Notice Board:**



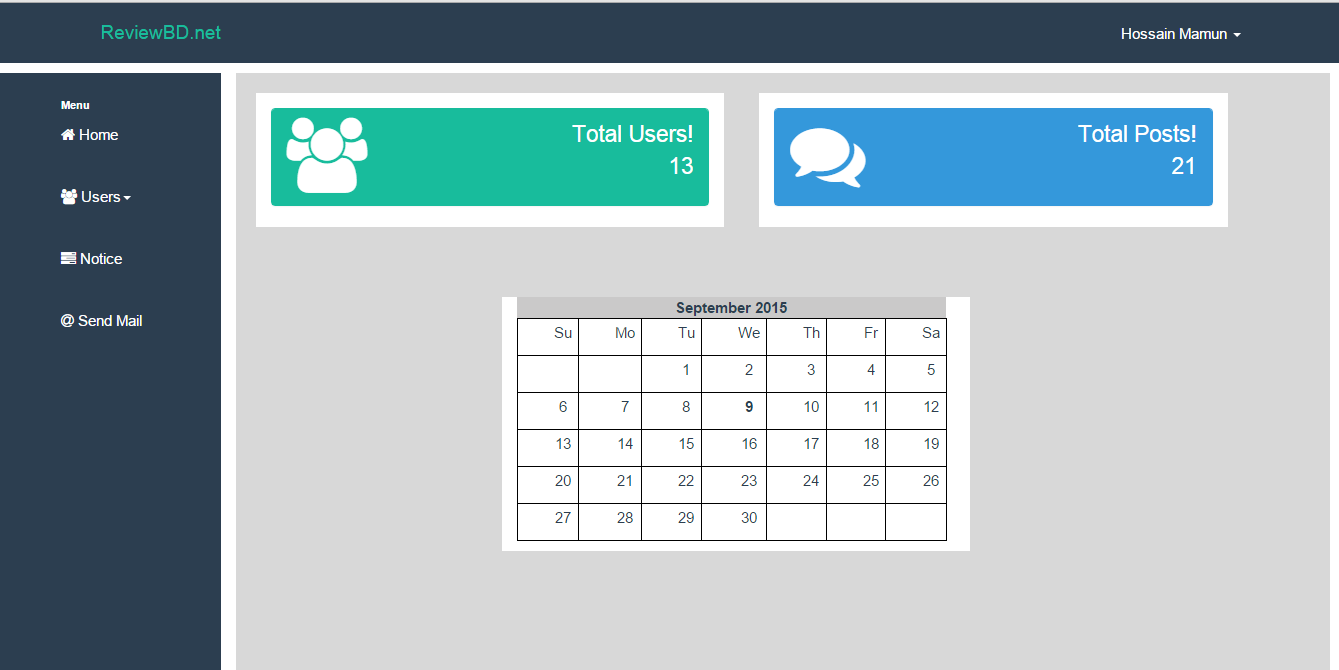
**Fig 4.11: Notice Board**

**4.2.12 Admin Sign in:**

****

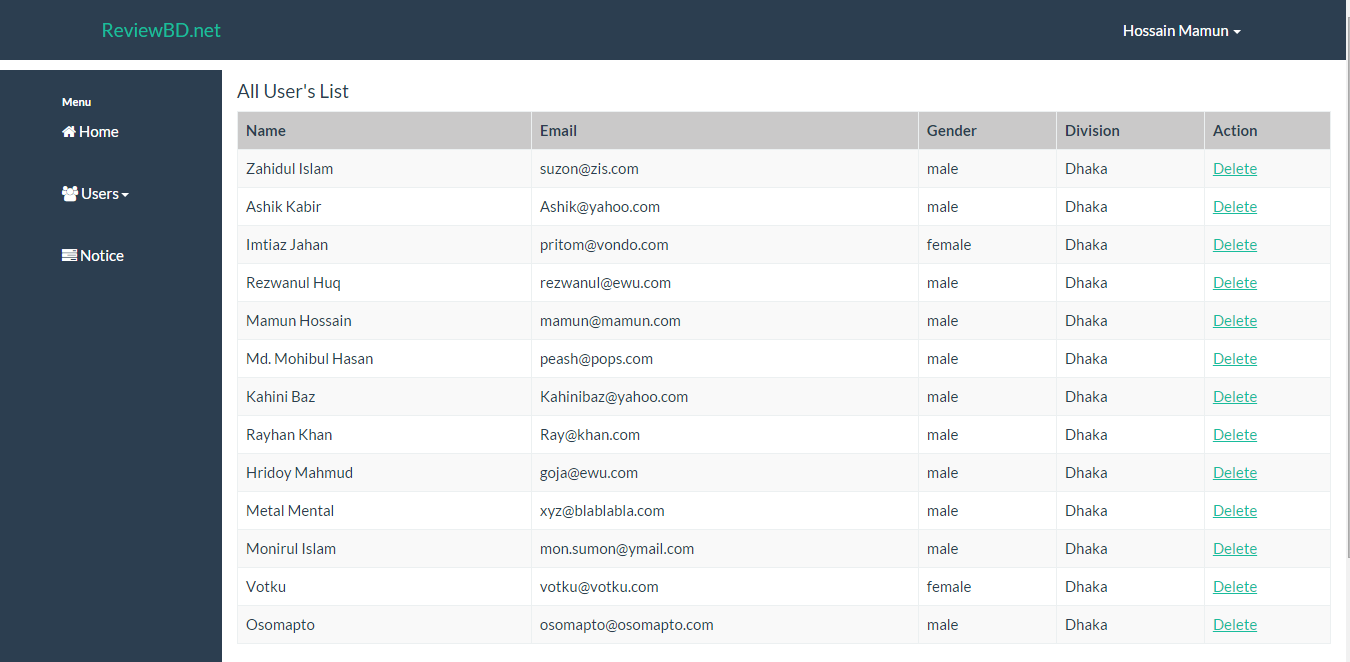
**Fig 4.12: Admin Sign in**

**4.2.13 Home:**

****

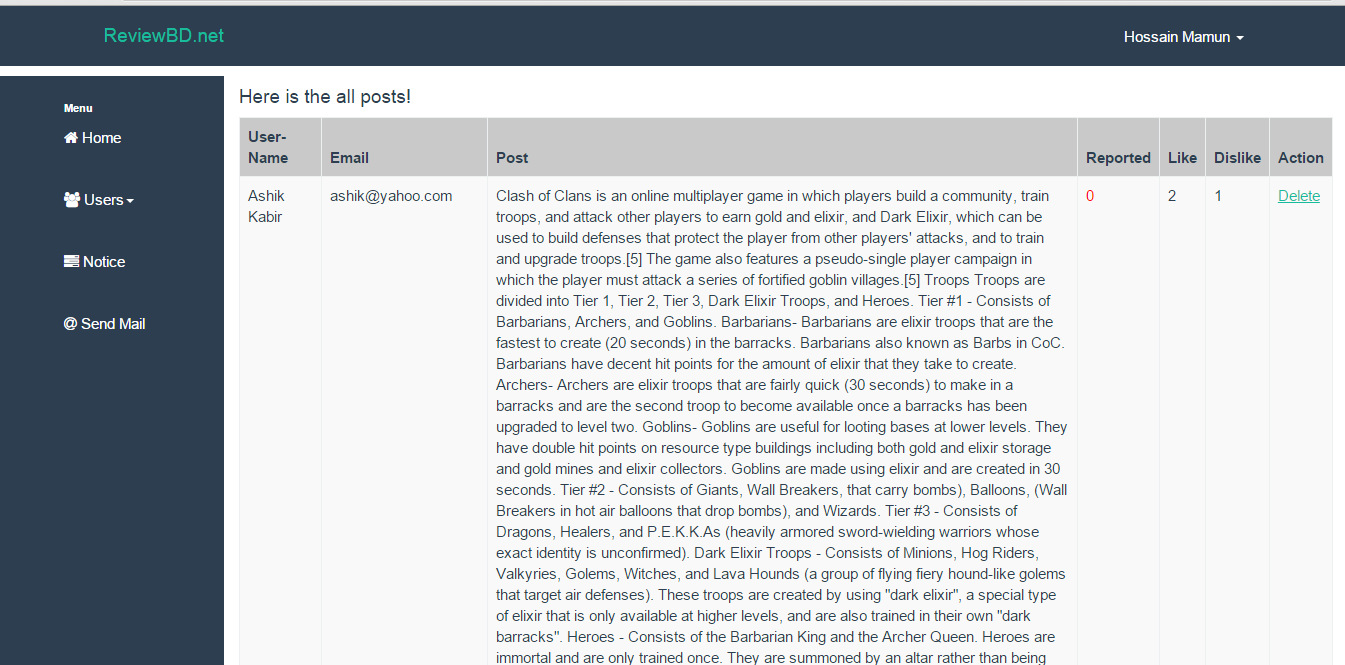
**Fig 4.13: Admin Home Page**

**4.2.14 Manage Users:**

**3**

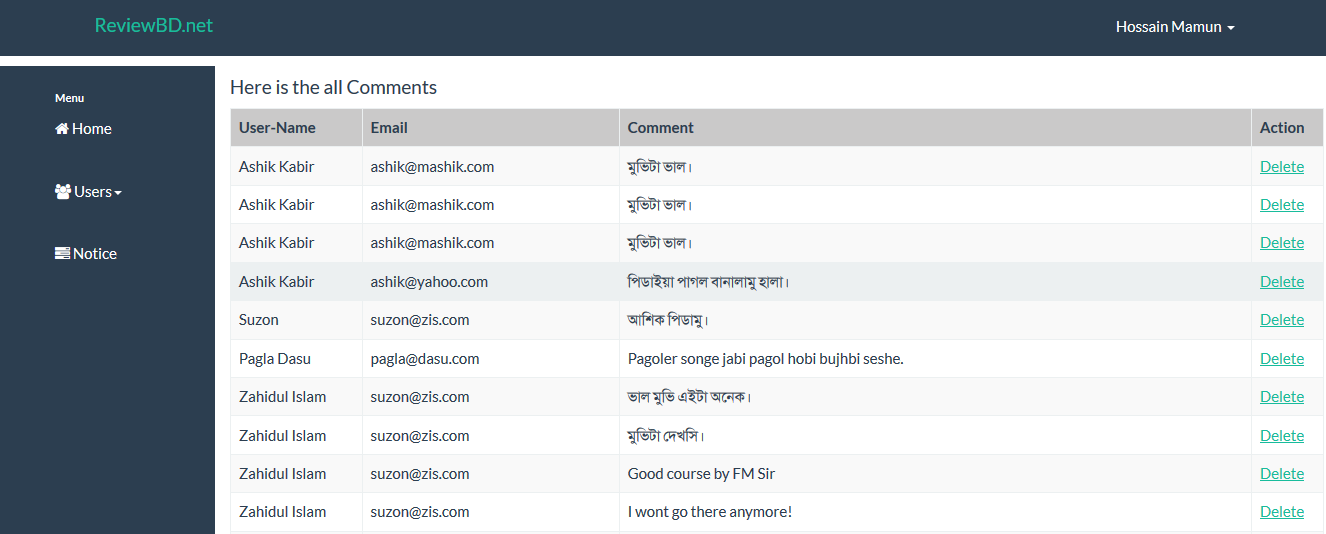
**Fig 4.14: Manage users**

**4.2.15 Manage Posts:**

****

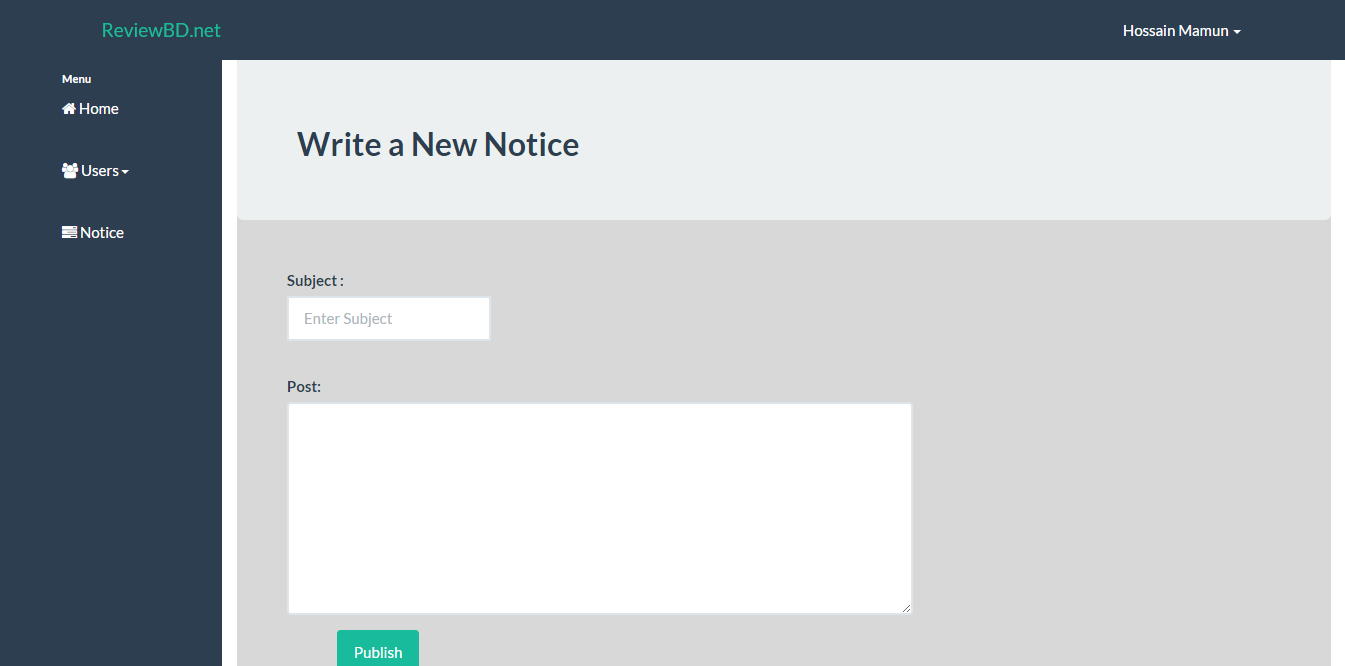
**Fig 4.15: Manage Posts**

**4.2.16 Manage Comments:**

****

**Fig 4.16: Manage Comments**

**4.2.17 Post Notice**



**Fig 4.17: Notice Post**

**Chapter 6**

**Future Work**

We will include lots of more categories based on user demands like book review and so on. Currently our database is not that dynamic, we will make it more efficient. However we will include email account verification. Because of ‘localhost’ SMTP protocol error, the email system is not working. We will include own rating system instead of **Facebook like API** so that we can use that to understand the user behavior using the data mining process. We will also build a monitoring system to see if a post is fraud or not.

**Chapter 7**

**Conclusion**

Bangladesh is a developing country. There are no online facilities to promote her own products by the consumers. We happen to see a lot of advertises of various products but they are busy in self-admiration. What is good what is bad is decided by the consumers. So if the consumers get the opportunity to tell what is good and what is bad, life would be so simple and the products companies will give more attention to make quality products. We are giving our users an online platform of “Word of Mouth” Marketing.

**Appendix**

In MVC framework we have Model for database, View for user view and Controller to control Model and View. Here we are showing the main function of the control.

**5.1 Home Page:**

public function index(){

$this->load->model('model\_users');

//--------------------Pagination----------------------

$this->load->library('pagination');

$config['base\_url'] = site\_url('site/index');

$config['total\_rows'] = $this->db->get('posts')->num\_rows();

$config['uri\_segment'] = 3;

$config['per\_page'] = 5;

$config['full\_tag\_open'] = "<ul class='pagination'>";

$config['full\_tag\_close'] ="</ul>";

$config['num\_tag\_open'] = '<li>';

$config['num\_tag\_close'] = '</li>';

$config['cur\_tag\_open'] = "<li class='disabled'><li class='active'><a href='#'>";

$config['cur\_tag\_close'] = "<span class='sr-only'></span></a></li>";

$config['next\_tag\_open'] = "<li>";

$config['next\_tagl\_close'] = "</li>";

$config['prev\_tag\_open'] = "<li>";

$config['prev\_tagl\_close'] = "</li>";

$config['first\_tag\_open'] = "<li>";

$config['first\_tagl\_close'] = "</li>";

$config['last\_tag\_open'] = "<li>";

$config['last\_tagl\_close'] = "</li>";

$this->pagination->initialize($config);

$pag = $this->db->order\_by("post\_id", "desc")->get('posts', $config['per\_page'], $this->uri->segment(3));

$table\_data\_home['allposts'] = $pag->result();

//------------end pagination-------------------------

/\*$table\_data\_home['allposts'] = $this->model\_users->get\_all\_posts();\*/

$this->load->view('my\_site',$table\_data\_home);

}

**5.2 Login Page:**

public function login()

{

$data['title']='Login';

$this->load->view('login/login',$data);

}

public function login\_validation()

{

$this->load->library('form\_validation');

$this->form\_validation->set\_rules('email','Email','required|trim|xss\_clean|callback\_validate\_credentials'); //xss prevent cross site scripting (hack er jonno use kore cross site scripting)

$this->form\_validation->set\_rules('password','Password','required|md5|trim');

if($this->form\_validation->run())

{

//Setting up the SESSION DATA

$mail = $this->input->post('email');

//$this->profile($mail);

$sql = "SELECT name from signup WHERE email = '".$mail."' LIMIT 1";

$result = $this->db->query($sql);

$row = $result->row();

$data=array(

'email' => $this->input->post('email'),

'name' => $row->name,

'is\_logged\_in' => 1

);

$this->session->set\_userdata($data);

redirect('site');

}

else{

$this->load->view('login/login');

}

}

**5.3 Signup:**

public function signup() //================>> SIGN UP

{

$this->load->view('signup/signup');

}

public function signup\_validation()

{

$this->load->library('form\_validation');

$this->form\_validation->set\_rules('email','Email','required|trim|valid\_email|is\_unique[signup.email]');

$this->form\_validation->set\_rules('password','Password','required|trim');

$this->form\_validation->set\_rules('cpassword','Confirm Password','required|trim|matches[password]');

$this->form\_validation->set\_message('is\_unique', 'The given email address already exits.');

if($this->form\_validation->run()){

$this->load->model('model\_users');

if($this->model\_users->add\_user())

{

/\*echo "hi ".$this->input->post('name')." Registred successfully" ;\*/

echo"<script language='javascript'>

alert('Registration Successfull! Please Login...');

window.location.href = 'login';

</script>

";

}

else

{

echo "Registration failed";

}

}else{

$this->load->view('signup/signup');

}

}

**5.4 Logout:**

public function logout(){

$this->session->sess\_destroy();

redirect('site');

}

**5.5 Profile:**

public function profile(){

if($this->session->userdata('is\_logged\_in')){

$this->load->model('model\_users');

$data = $this->model\_users->show\_pp();

//print('/uploads/'.$data);

$img = base\_url().$data;

$table\_data['allposts'] = $this->model\_users->get\_user\_posts();

$pass\_the\_img = array('img' => $img);

$data = array\_merge($pass\_the\_img, $table\_data);

$this->load->view('profile/profile', $data);

}

else{

echo "You are not authorized...";

}

}

public function save\_pp(){

$url = $this->do\_upload();

$this->load->model('model\_users');

if(!$this->model\_users->save\_pp\_model($url))

{

echo "Uploaded successfully" ;

redirect('site/profile');

}

else

{

echo "Upload failed";

}

}

private function do\_upload(){

$type = explode('.', $\_FILES["pic"]["name"]);

$type = $type[count($type)-1];

$url = "./images/pp/".uniqid(rand()).'.'.$type;

if(in\_array($type, array("jpg","jpeg","gif","png"))){

if(is\_uploaded\_file($\_FILES["pic"]["tmp\_name"])){

if(move\_uploaded\_file($\_FILES["pic"]["tmp\_name"], $url))

{

return $url;

}

}

return "";

}

}

function show\_pp()

{

$this->load->model('model\_users');

$data = $this->show\_pp();

}

**5.6 Write New Review:**

public function write\_new\_blog(){

if($this->session->userdata('is\_logged\_in')){

$this->load->view('newblog/newblog');

}

else echo "You are not authorized to see this page. Please login! ";

}

public function post\_new\_blog(){

$this->load->library('form\_validation');

$this->form\_validation->set\_rules('blog\_post','Post','required');

$this->form\_validation->set\_rules('blog\_title','Title','required|trim');

$this->form\_validation->set\_rules('tag','Tag','required');

$this->form\_validation->set\_rules('categories','Categories','required');

if($this->form\_validation->run()){

$post\_img\_url = $this->post\_img\_do\_upload();

$this->load->model('model\_users');

if($this->model\_users->write\_new\_post($post\_img\_url))

{

echo " Your post has been published successfully" ;

redirect('site/profile/profile');

}

}

else

{

$this->load->view('newblog/newblog');

}

}

private function post\_img\_do\_upload(){

$type = explode('.', $\_FILES["pic"]["name"]);

$type = $type[count($type)-1];

$url = "./images/postimgs/".uniqid(rand()).'.'.$type;

if(in\_array($type, array("jpg","jpeg","gif","png"))){

if(is\_uploaded\_file($\_FILES["pic"]["tmp\_name"])){

if(move\_uploaded\_file($\_FILES["pic"]["tmp\_name"], $url))

{

return $url;

}

}

return "";

}

}

**5.7 Categorical Post:**

public function categorial\_post(){

$data['category\_name'] = $\_GET['category\_name'];

$for\_categort\_page\_title = $\_GET['category\_name'];

$this->load->model('model\_users');

$table\_data\_category['allposts'] = $this->model\_users->model\_for\_category\_post($data);

$this->load->view('category\_post/categorial\_post', $table\_data\_category);

}

**5.8 Viewing a complete post:**

public function full\_post(){

$data['postID'] = $\_GET['post\_id']; // home page theke URL er maddhome $post\_id variable ta nie aschi

$this->load->model('model\_users');

$whole\_single\_post['single\_post'] = $this->model\_users->get\_whole\_single\_post($data);

$sub\_categories['sub\_categories'] = $this->model\_users->get\_sub\_categories($data);

$post\_comments['all\_comments'] = $this->model\_users->get\_all\_comments($data);

$pass\_the\_array\_of\_post\_and\_comments = array\_merge($whole\_single\_post, $post\_comments, $sub\_categories, $data);

$this->load->view('post\_view/post',$pass\_the\_array\_of\_post\_and\_comments);

}

**5.9 Comment:**

public function publish\_comments(){

$this->load->model('model\_users');

$this->model\_users->insert\_comment();

}

**5.10 Report Count:**

public function report\_count(){

$post\_id = $\_POST['report'];

$value = 1;

/\*echo $post\_id ; echo $value;\*/

echo"<script language='javascript'>

alert('Post Reported');

window.history.go(-1);

</script>

";

$this->load->model('model\_users');

$this->model\_users->update\_post\_report($value,$post\_id);

}

**5.11 Search:**

public function search(){

$search\_words = $\_POST['search'];

$search\_words = trim($search\_words);

$spliting\_search\_words = explode(" ", $search\_words);

$this->load->model('model\_users');

$table\_data\_search['search\_result'] = $this->model\_users->search\_function($spliting\_search\_words);

$this->load->view('search/search', $table\_data\_search);

}

**5.12 Notice Board:**

public function notice\_board(){

$this->load->model('model\_users');

$notice['notice\_post'] = $this->model\_users->get\_notice();

$this->load->view('notice/notice', $notice);

}

**5.13 Home (Admin Panel):**

public function index(){

$this->load->model('admincheck\_model');

if($this->admincheck\_model->is\_admin\_logged\_in()){

$total\_user=$this->admincheck\_model->user\_count();

$total\_post=$this->admincheck\_model->post\_count();

$data['cal']=$this->calendar->generate();

$data['homepage']='homepage';

$data['home']='Home';

$data['userlist']=$total\_user;

$data['postcount']=$total\_post;

$this->load->view('dashboard\_layout',$data);

}

else{

$this->load->view('admin\_view');

}

}

public function validation(){

$this->form\_validation->set\_rules('email', 'Email', 'required|valid\_email');

$this->form\_validation->set\_rules('pword', 'Password', 'required');

if ($this->form\_validation->run() == FALSE)

{

$this->load->view('admin\_view');

}

else

{

$result=$this->admincheck\_model->admin\_check();

if($result){

redirect('dashboard\_con');

}

else{

$data=array();

$data['error\_login']='Username or Password is invalid!';

$this->load->view('admin\_view',$data);

}

}

}

**5.14 Logout (Admin Panel):**

public function logout(){

if($this->admincheck\_model->is\_admin\_logged\_in()){

$this->session->unset\_userdata('current\_admin\_id');

$this->session->unset\_userdata('current\_admin\_name');

$this->session->sess\_destroy();

redirect('admin\_con/?logout=success');

}

else{

$this->load->view('admin\_view');

}

}

**5.15 All Users (Admin Panel):**

public function all\_users(){

$this->load->model('admincheck\_model');

$data['allUsers']='allUsers';

$data['manage\_users']='Users List';

$data['allUserData']= $this->admincheck\_model->all\_user\_list();

$this->load->view('dashboard\_layout',$data);

}

**5.16 Delete User (Admin Panel):**

public function user\_delete($id){

if ($this->admincheck\_model->user\_delete($id)) {

redirect('dashboard\_con/all\_users');

}

else{

redirect('dashboard\_con');

}

}

**5.17 All Posts (Admin Panel):**

public function all\_posts(){

$this->load->model('admincheck\_model');

$data['allPosts']='allPosts';

$data['user\_post']='Users Post';

$data['allUserPost']= $this->admincheck\_model->all\_post\_list();

$this->load->view('dashboard\_layout',$data);

}

#post delete

public function post\_delete($id){

if ($this->admincheck\_model->post\_delete($id)) {

redirect('dashboard\_con/all\_posts');

}

else{

redirect('dashboard\_con');

}

}

**5.18 All comments (Admin Panel):**

#all Comments

public function all\_comments(){

$this->load->model('admincheck\_model');

$data['allComments']='allComments';

$data['user\_comment']='Users Comment';

$data['allUserComment']= $this->admincheck\_model->all\_comment\_list();

$this->load->view('dashboard\_layout',$data);

}

#comment delete

public function comment\_delete($id){

if ($this->admincheck\_model->comment\_delete($id)) {

redirect('dashboard\_con/all\_comments');

}

else{

redirect('dashboard\_con');

}

}

**5.19 Notice Board:**

public function notice(){

$data['Notice']='Notice';

$data['notice']='Write New Notice';

$this->load->view('dashboard\_layout',$data);

}

public function notice\_validation(){

$this->form\_validation->set\_rules('subject', 'Subject', 'required');

$this->form\_validation->set\_rules('notice\_post', 'Text', 'required');

if ($this->form\_validation->run() == FALSE)

{

$data['Notice']='Notice';

$data['notice']='Wrong Validity';

$this->load->view('dashboard\_layout',$data);

}

else{

if ($this->admincheck\_model->notice\_insert()) {

$data['Notice']='Notice';

$data['notice']='Success Message';

$data['success']='Your notice has been written successfully!';

$this->load->view('dashboard\_layout',$data);

}

}

}

}

**Reference**

[1] http://www.w3schools.com/htmL/html\_basic.asp, July 20, 2015

[2] <http://www.w3schools.com/css/css_intro.asp>, July 22, 2015

[3] <http://getbootstrap.com/>, August 01, 2015

[4] <http://www.w3schools.com/js/default.asp>, August 05, 2015

[5] <http://www.w3schools.com/jquery/default.asp>, August 08, 2015

[6] http://www.php.net, PHP: Hypertext Preprocessor, August 15, 2015

[7] <http://www.codeigniter.com/user_guide/general/controllers.html>, August 18, 2015

[8] <https://en.wikipedia.org/wiki/MySQL>, July 30, 2015