Secure Internet Application Development

CPS 499-63/592-65, Summer 2017

Final Project :: Integrated Chat based Blog System

Designing and Implementation of Blog System:

It is a Web based Secured-Chat integrated Blog system. It is two-level heirarchical system.

The two levels are as follows:

1. Guest Level

Guest can only view the posts and add his comments
It is secured based blog system as we have employed few security measures into it. How my blog

- Login:
  log.php

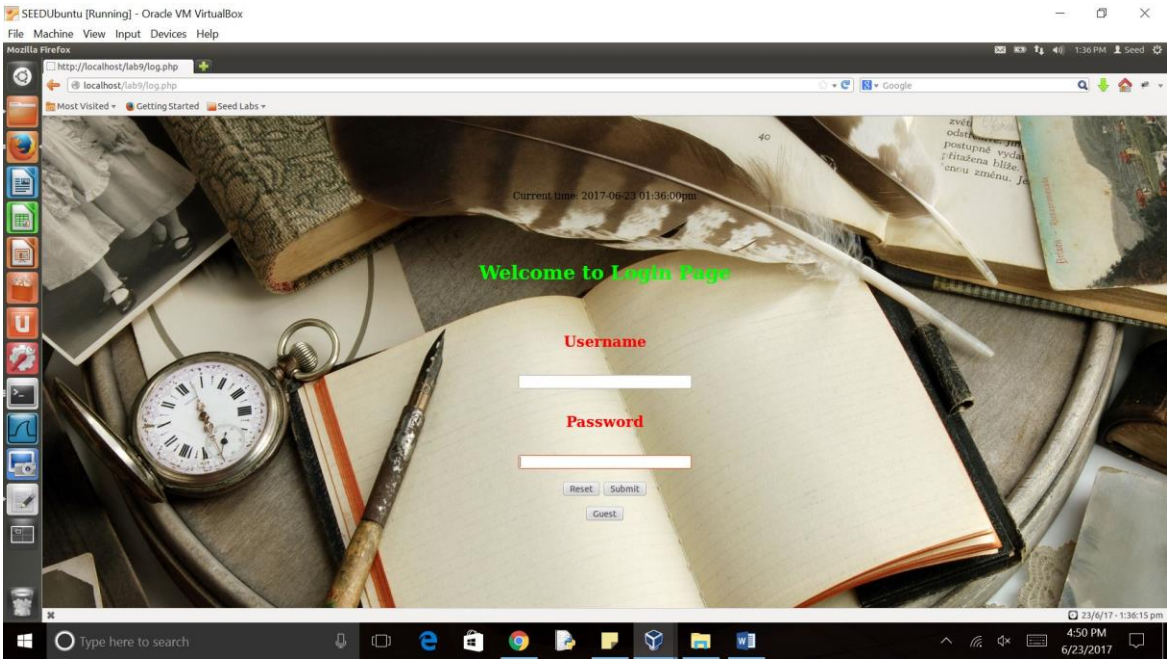


Figure 1: Login page (log.php)

- Guest:

guest.php

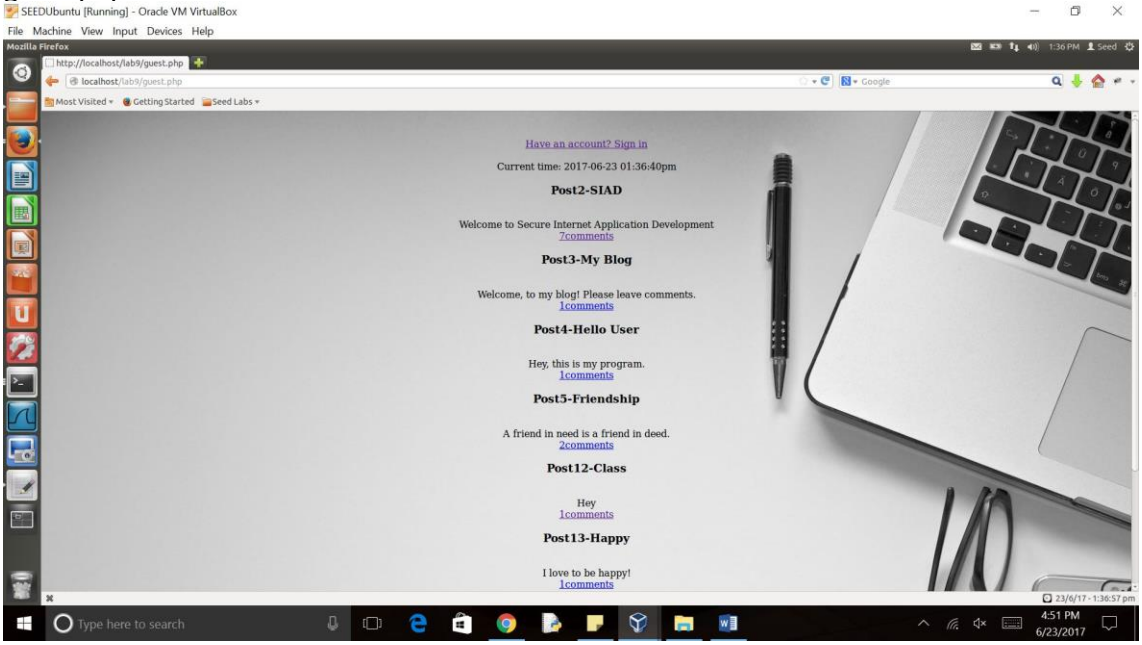


Figure 2: Guest-welcome page (guest.php)

View Comments: guestcomment.php

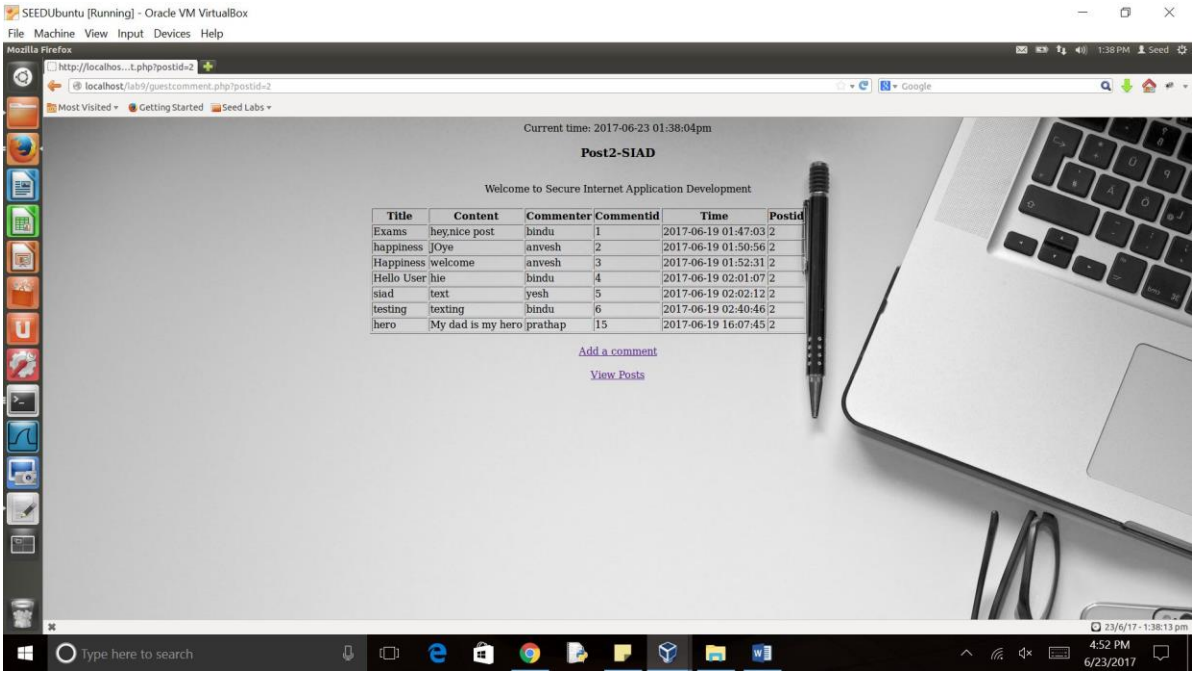


Figure 3: Guest- view comments (guestcomment.php)

Add Comments: guestcform.php

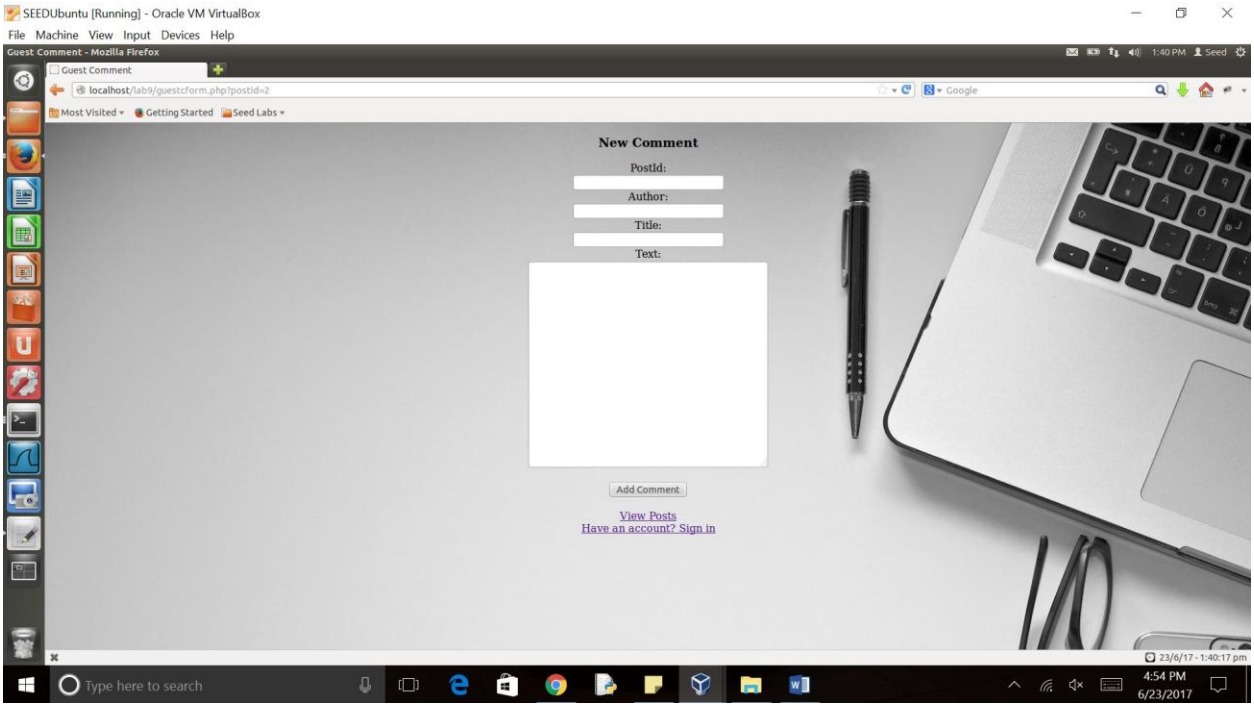


Figure 4: Guest- Add Comment (guestcform.php)

2. User Level

Authentication is required. User can view the posts, edit and delete them. He can scribe comments, edit and delete the comments too.
If user needs any help, he can chat with online users instantly. Chat feature: He can view available online users and chat.

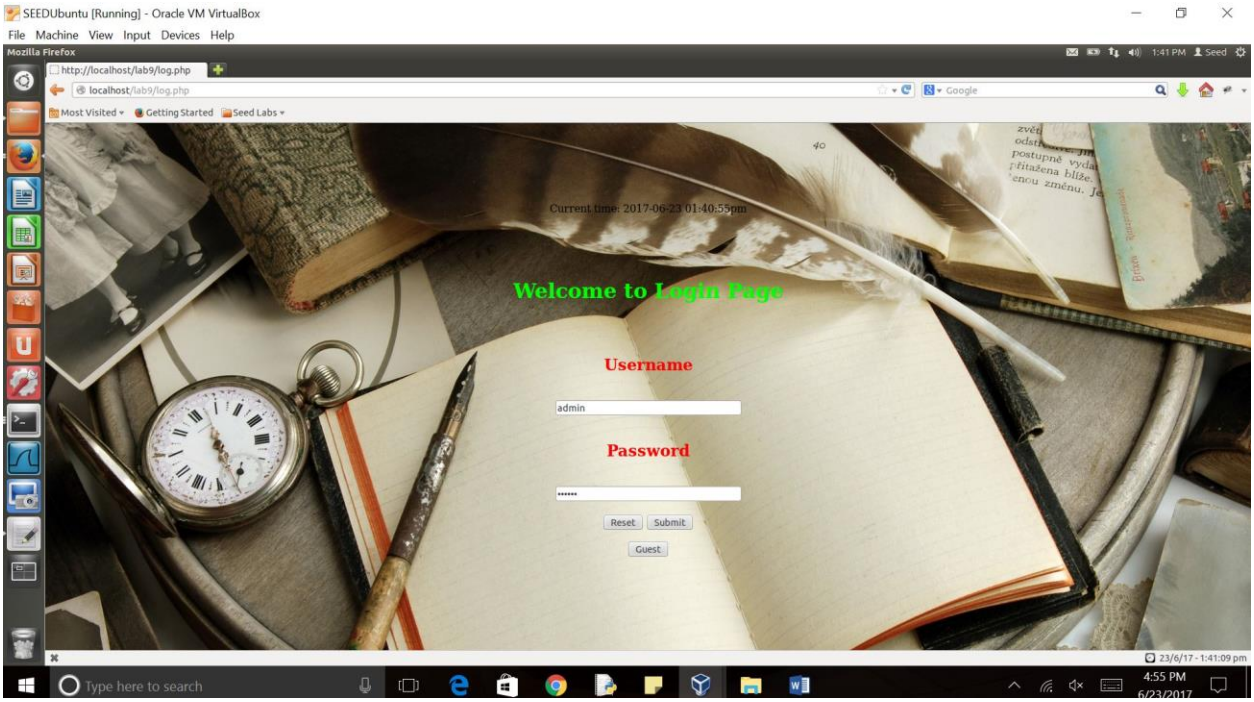


Figure 5: Login with valid credentials

Welcome Page:
(index.php)

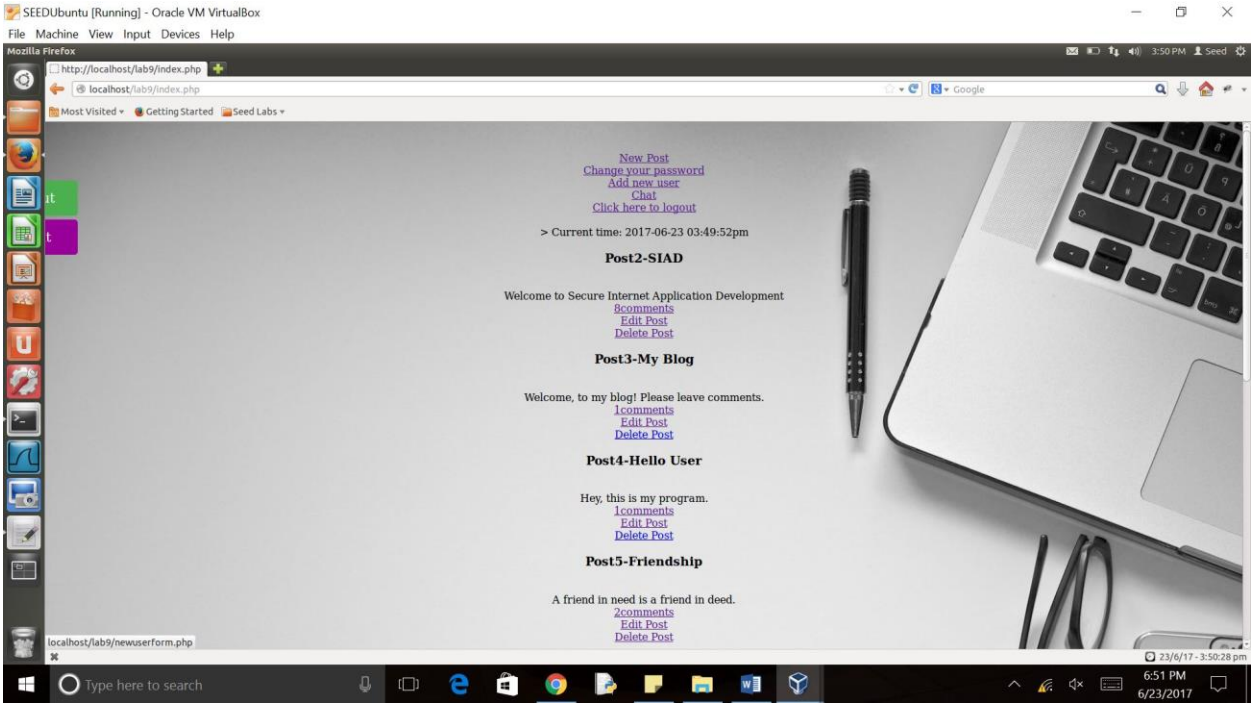


Figure 6 : Welcome page

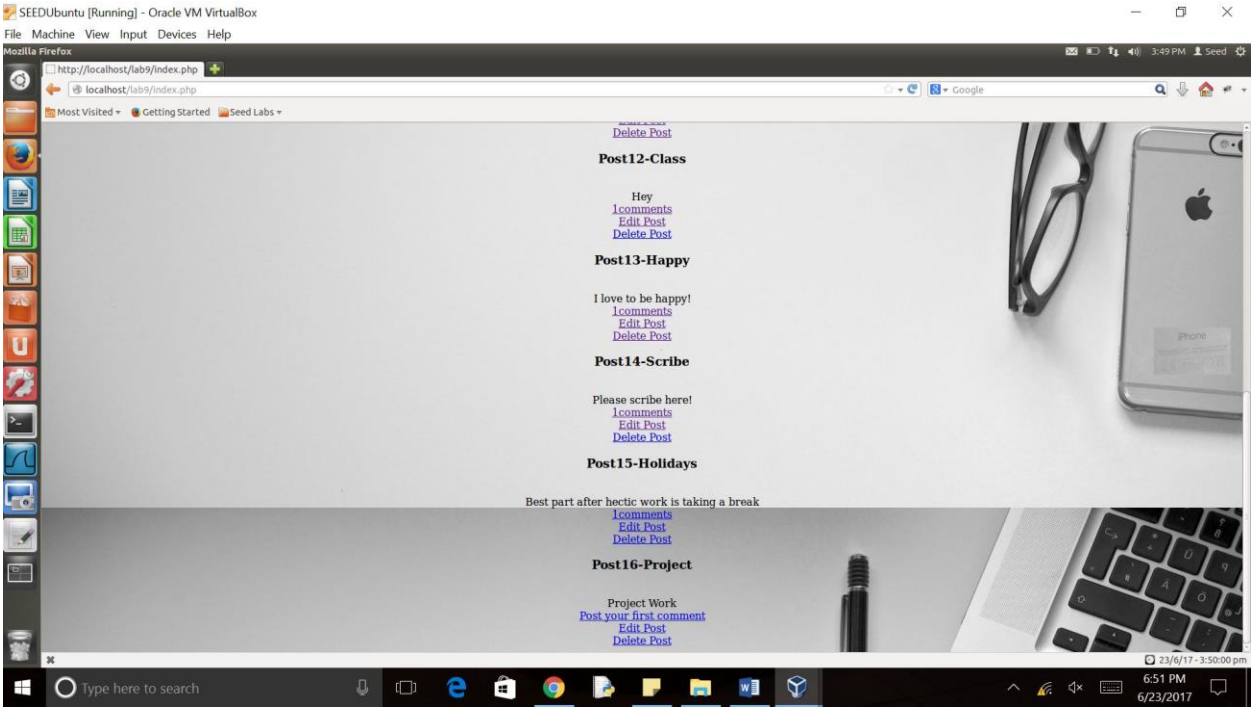


Figure 7 : Welcome page 1.1

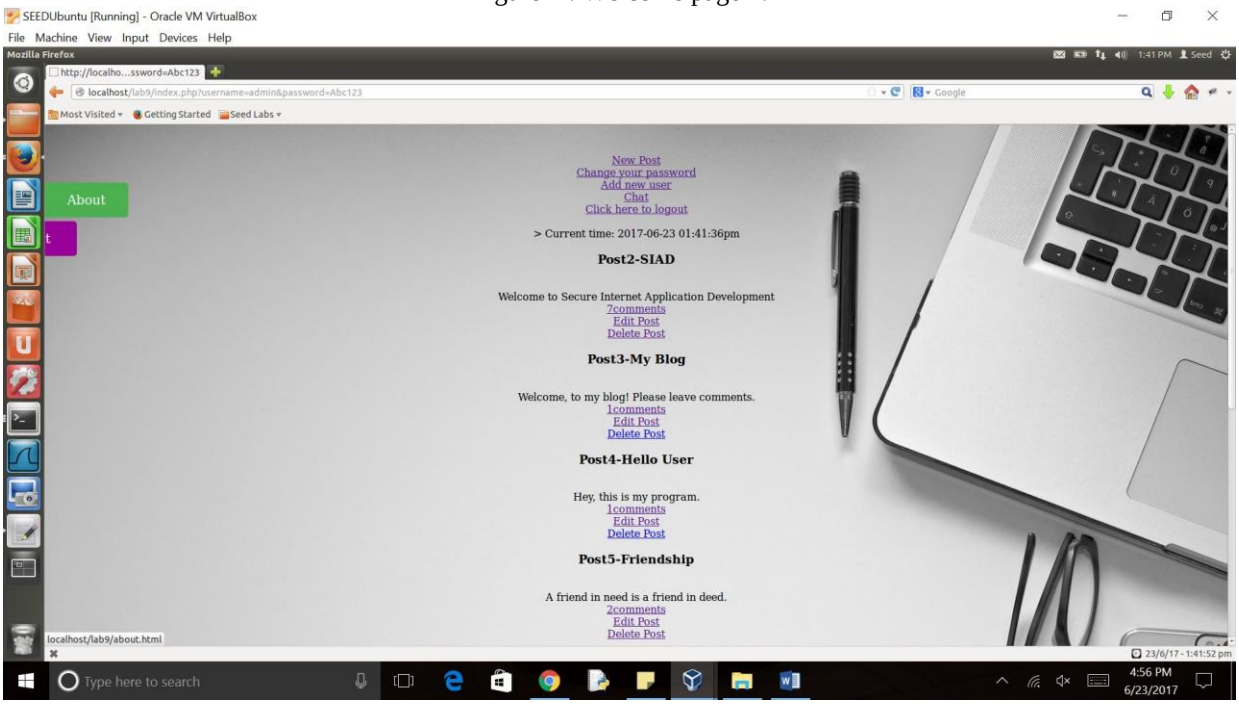


Figure 8:Welcome page 1.2

About Page:
(about.html)

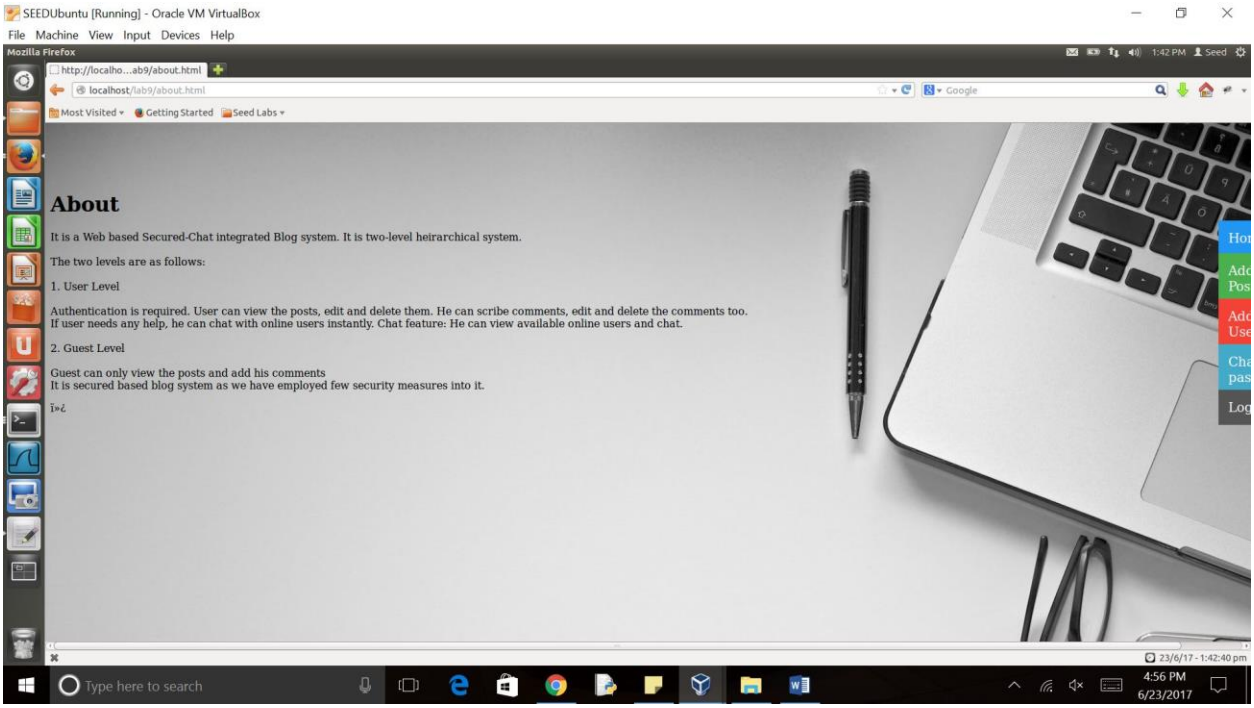


Figure 9: About page

Chat:
(websocketchat1.php)

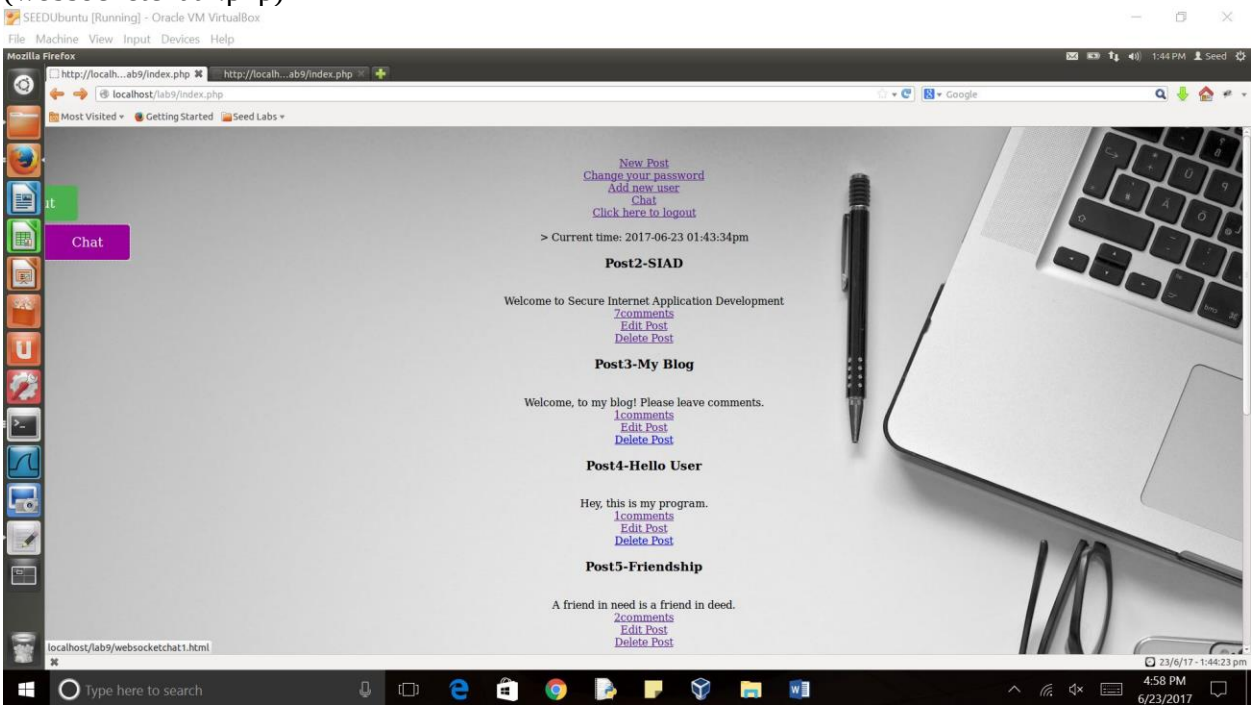


Figure 10: Chat

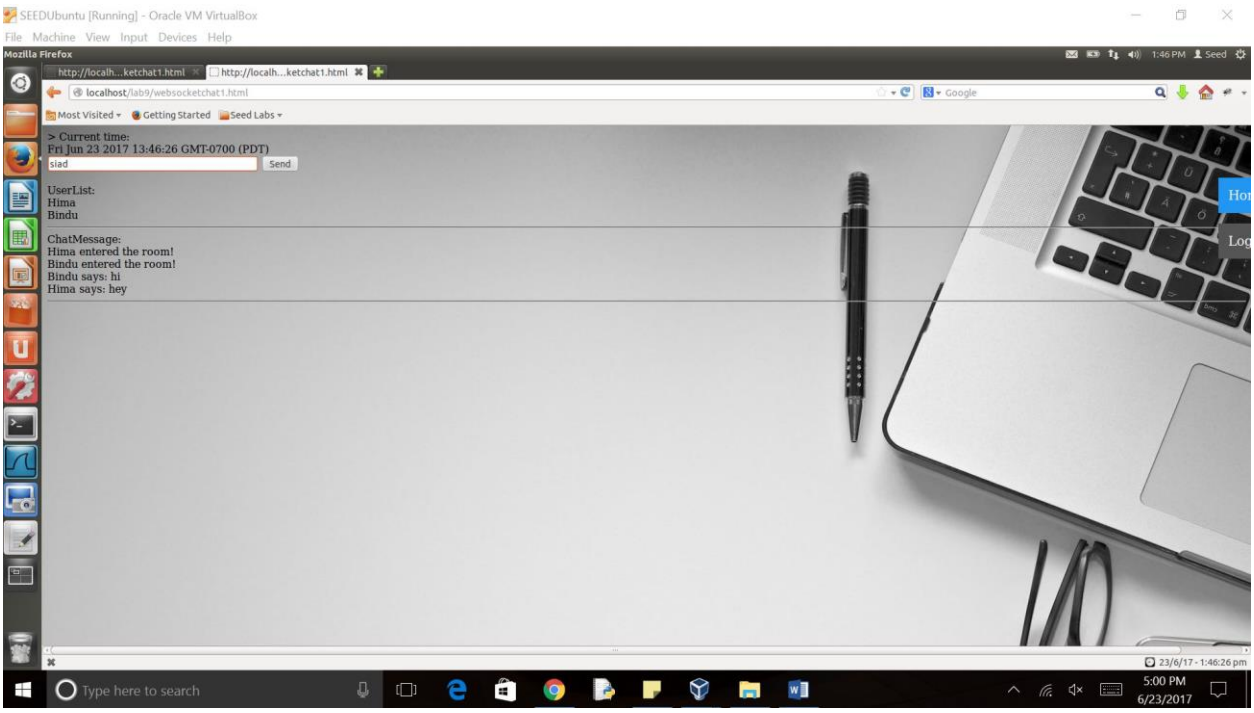


Figure 11: Chat

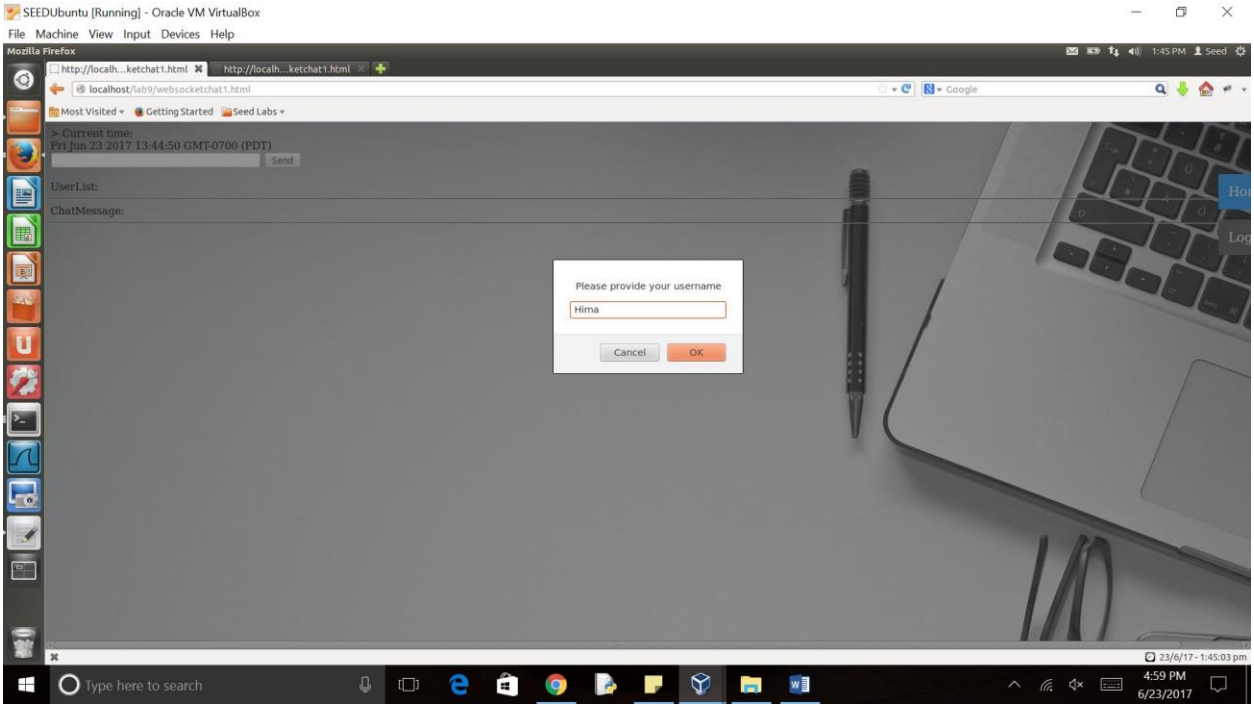


Figure 12 Chat- enter name

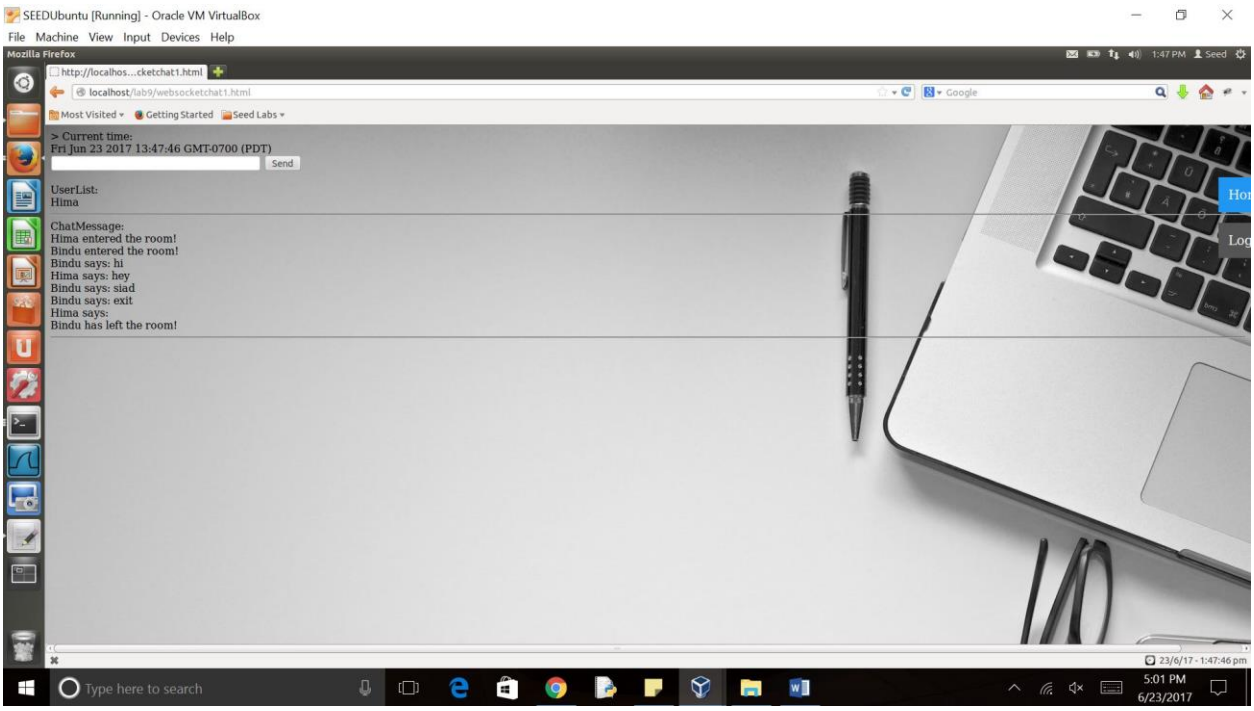


Figure 13: Chat close

Change Password Form:

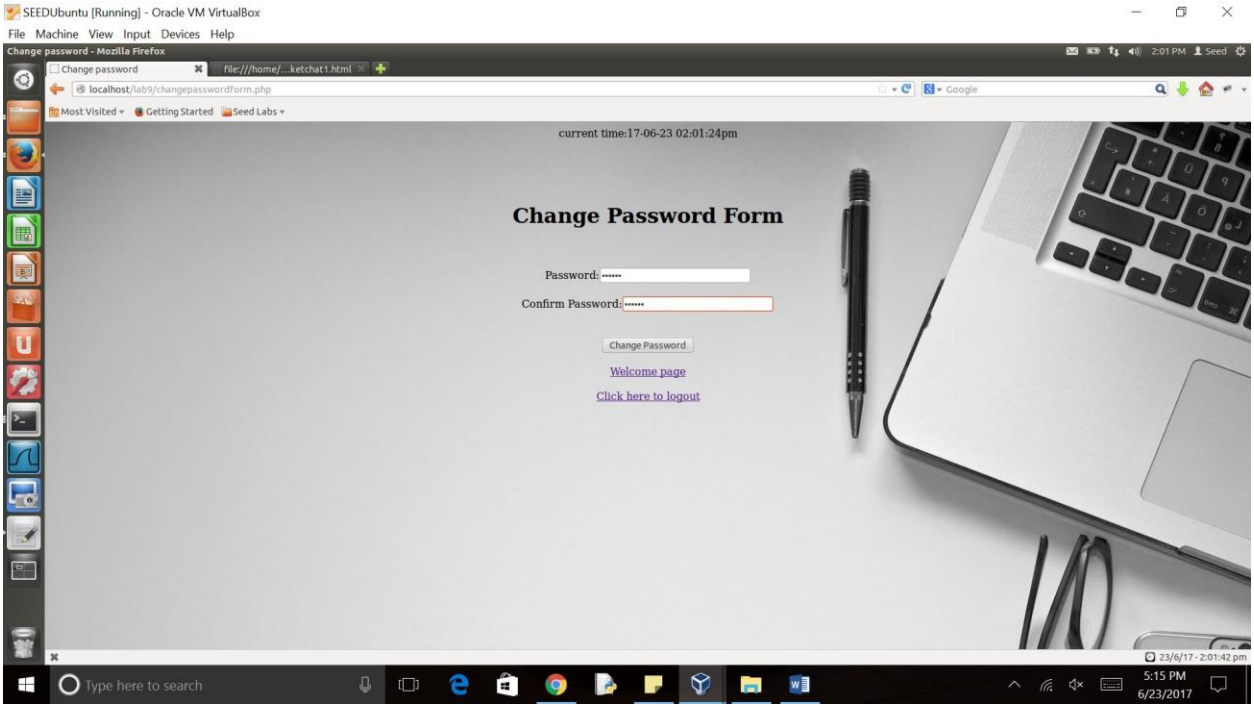


Figure 14: Change Password Form

Add Post :

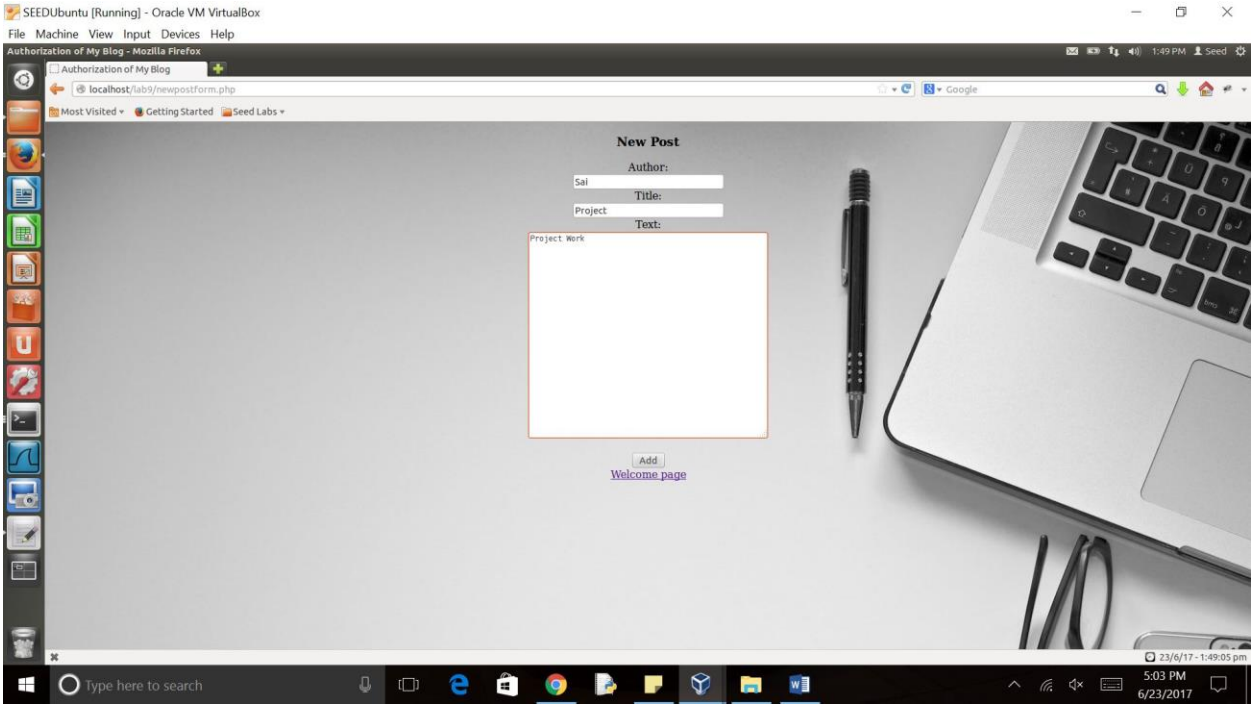
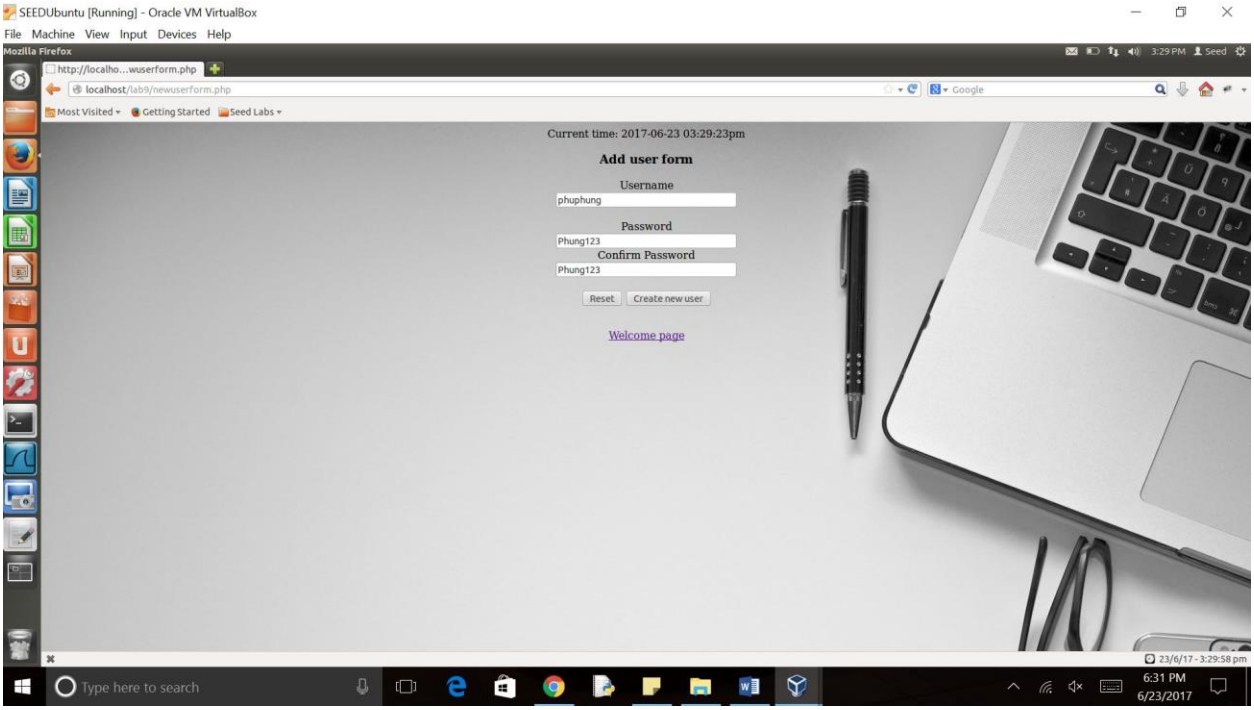
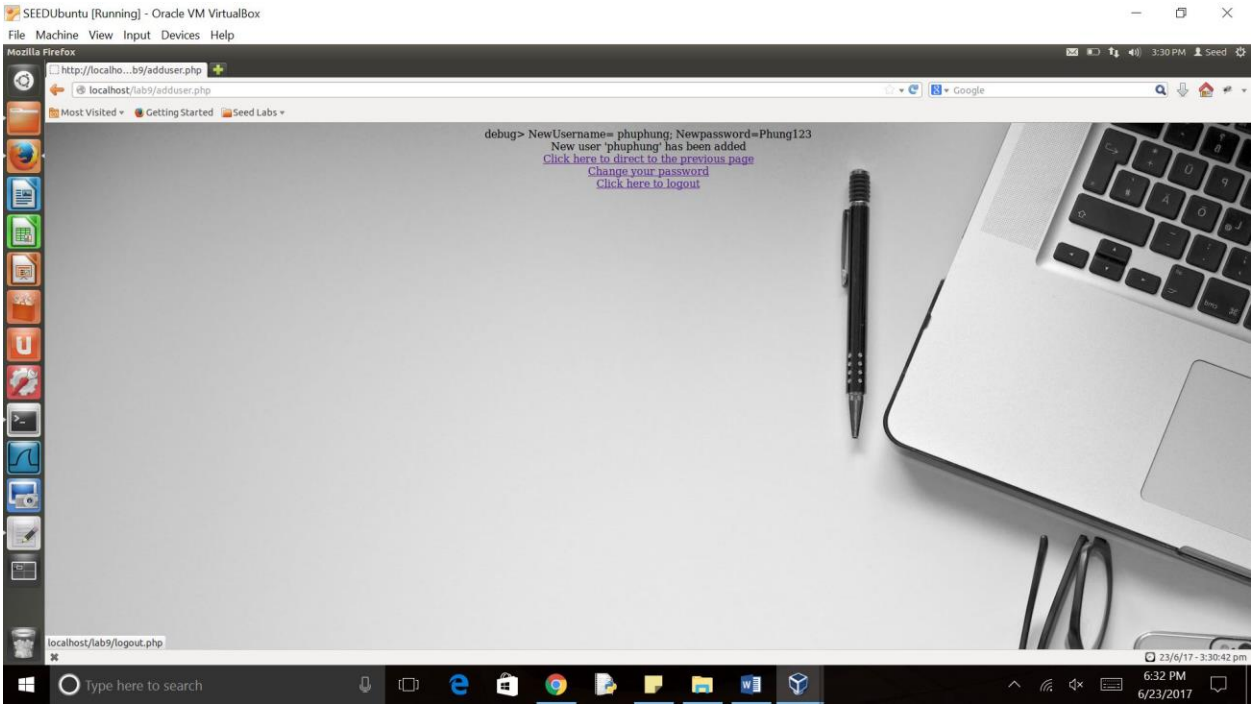


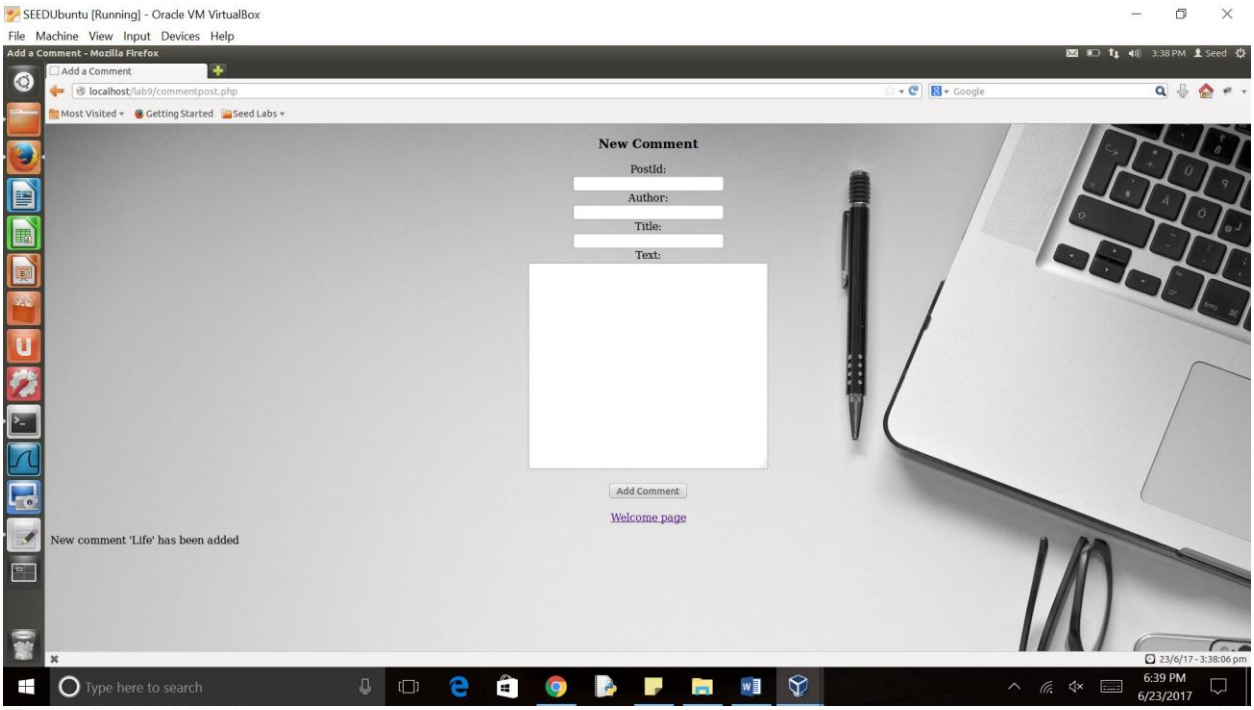
Figure 15: Add new post

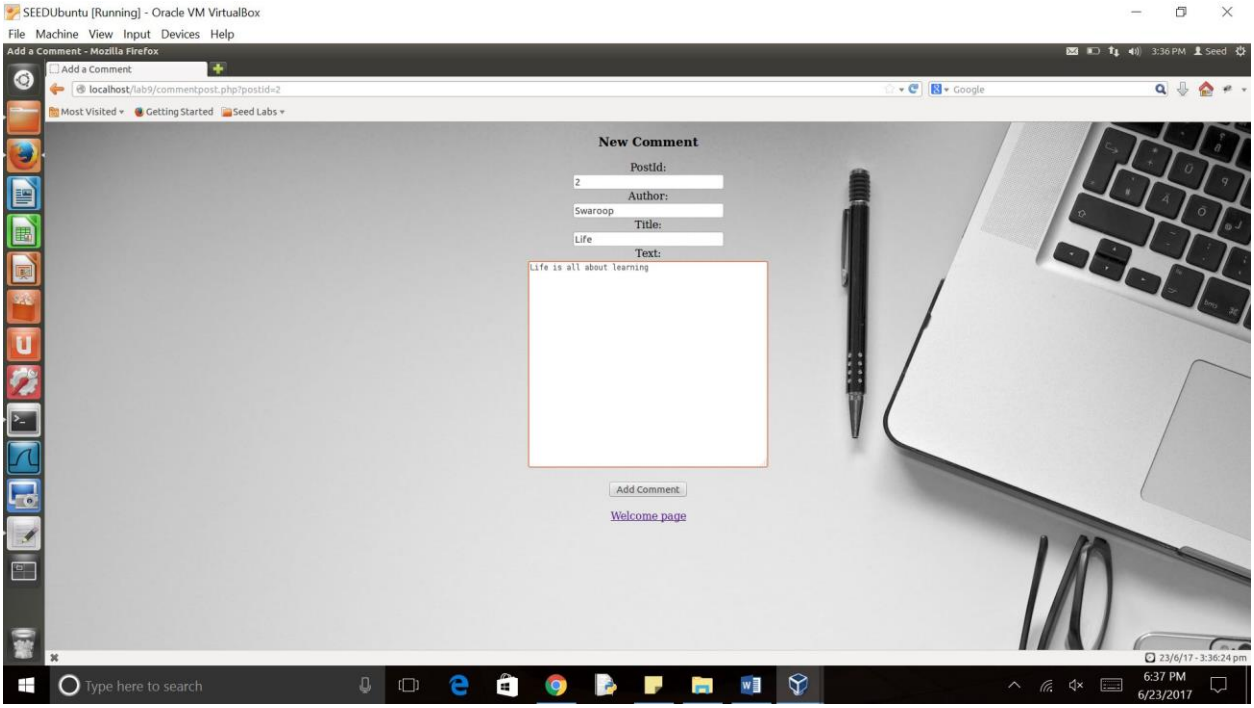
Add user form



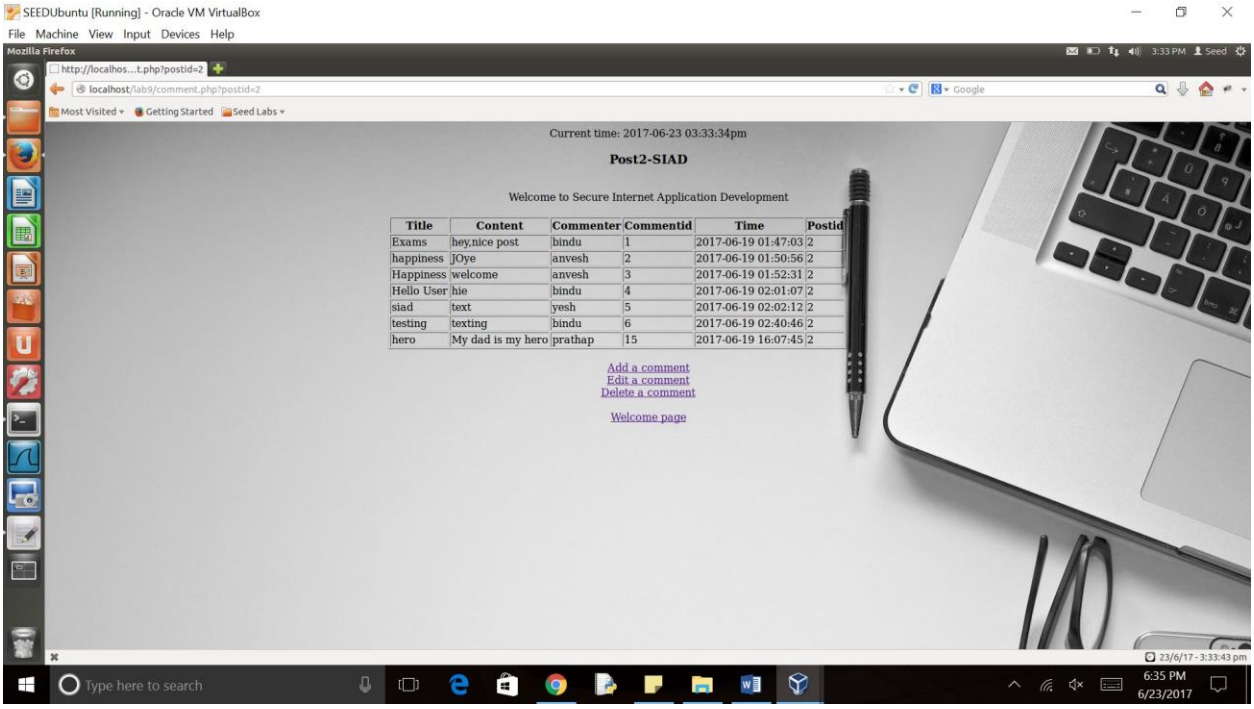

SEEDUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
http://localho...b9/adduser.php
localhost/lab9/adduser.php
Most Visited
Getting Started
Seed Labs
debug> NewUsername= phuphung; Newpassword=Phung123
New user 'phuphung' has been added
Click here to direct to the previous page
Change your password
Click here to logout
localhost/lab9/logout.php
Type here to search
6:32 PM
6/23/2017

Add Comment form

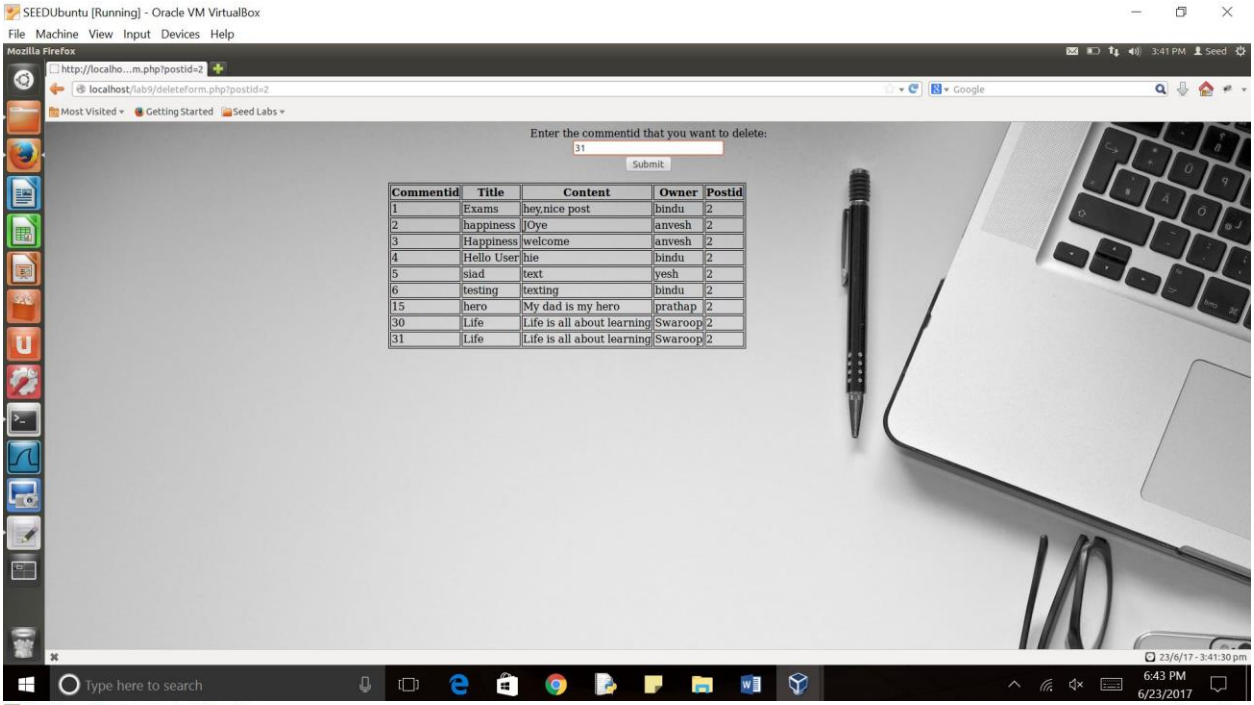


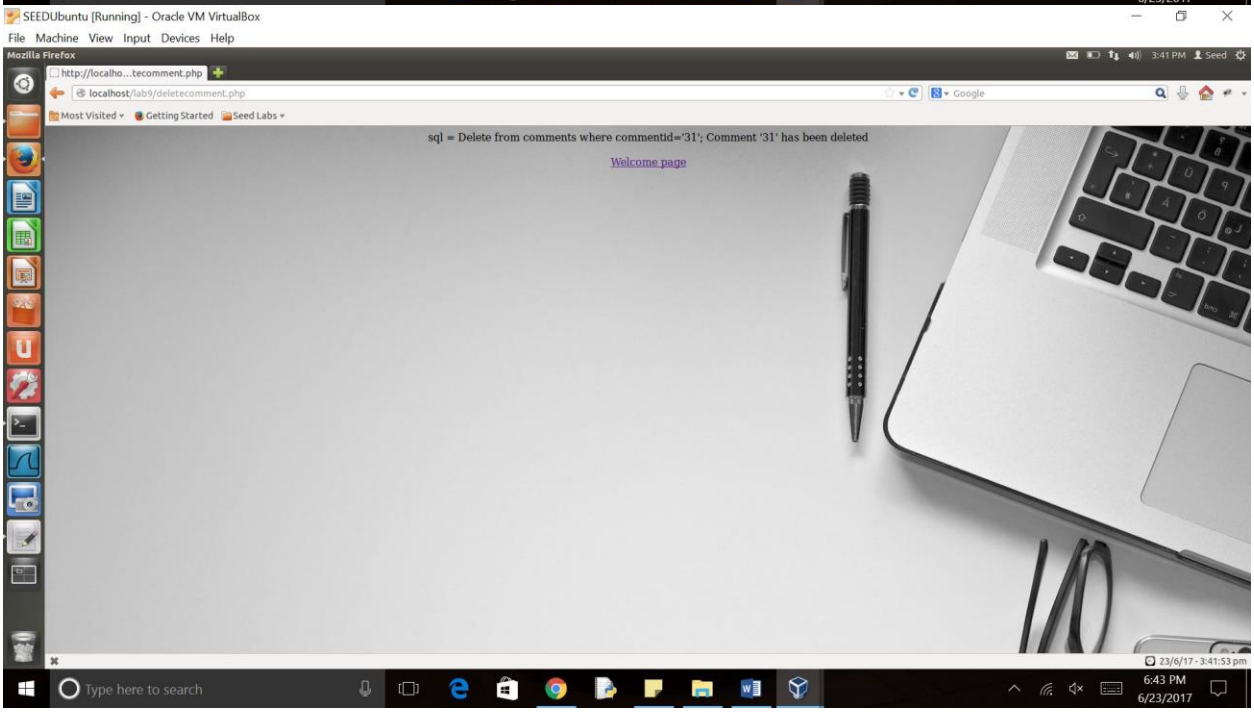


View Comments:

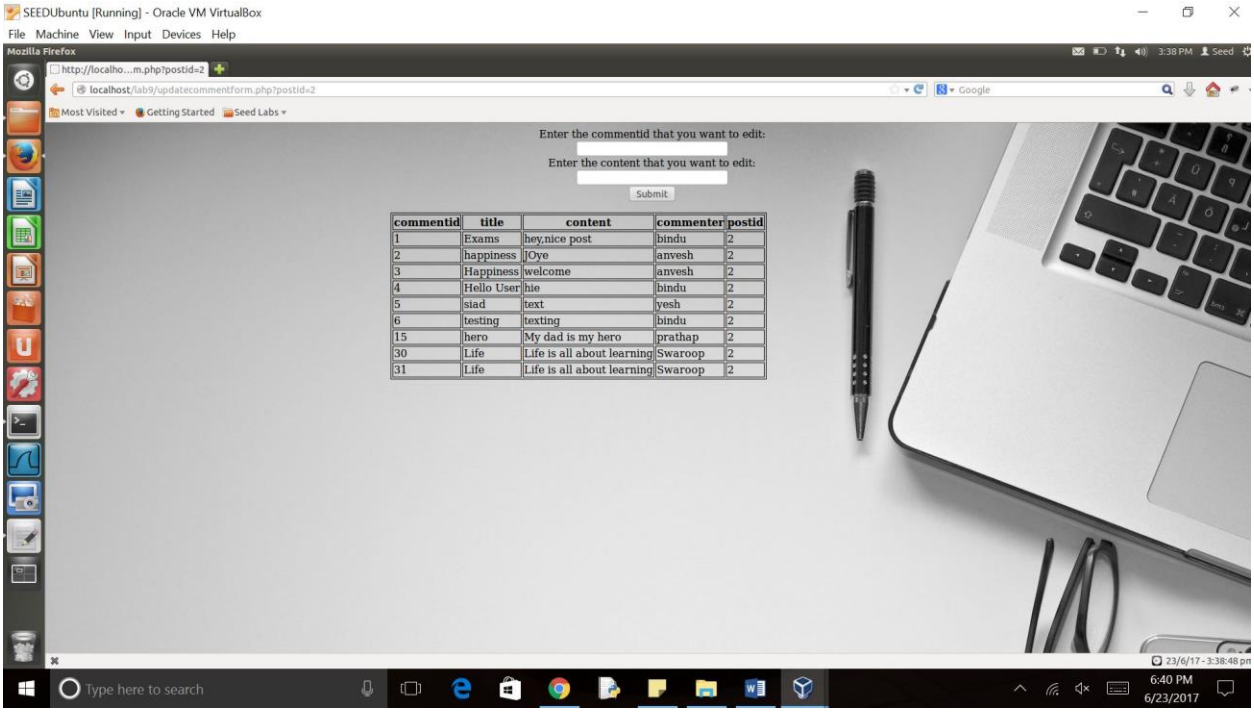


Delete Comments

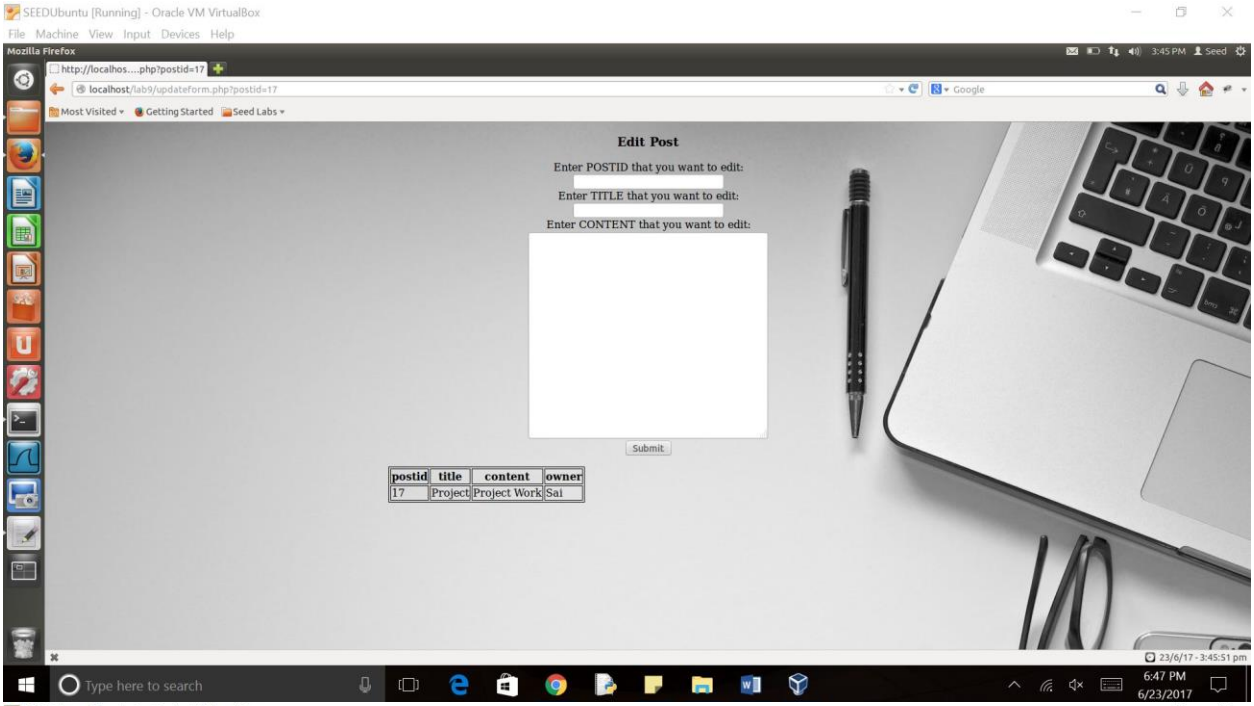


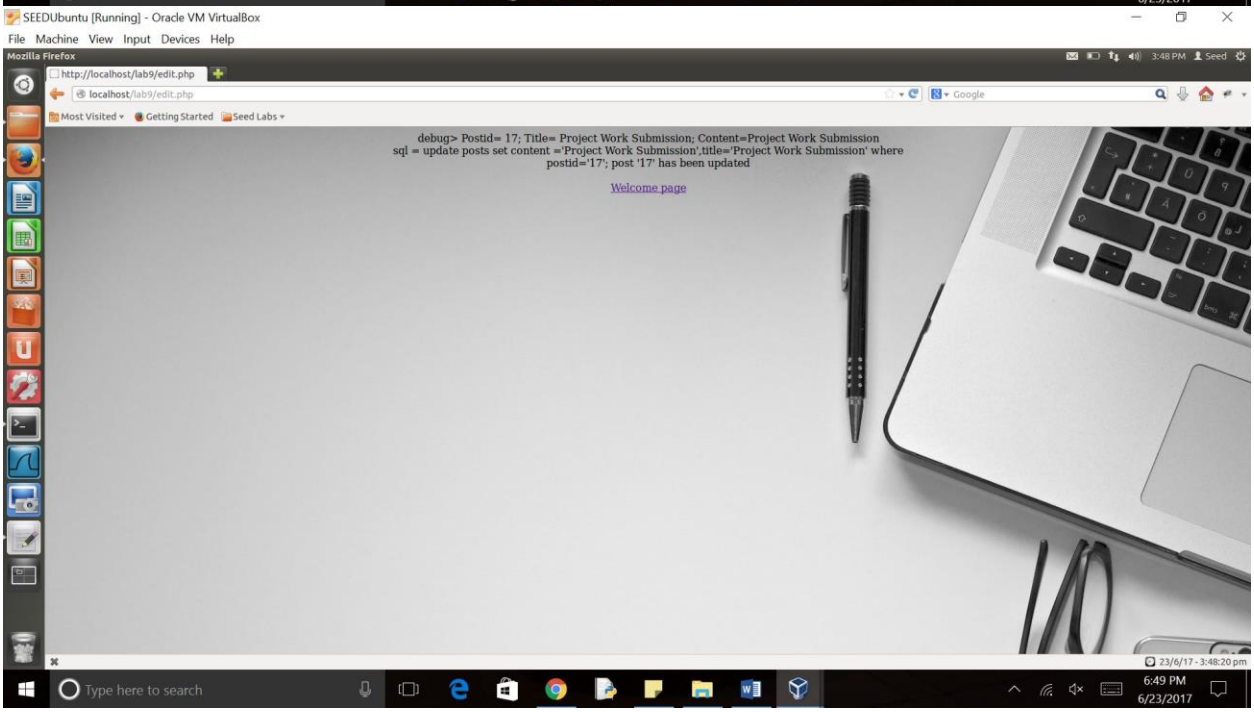


Edit Comments

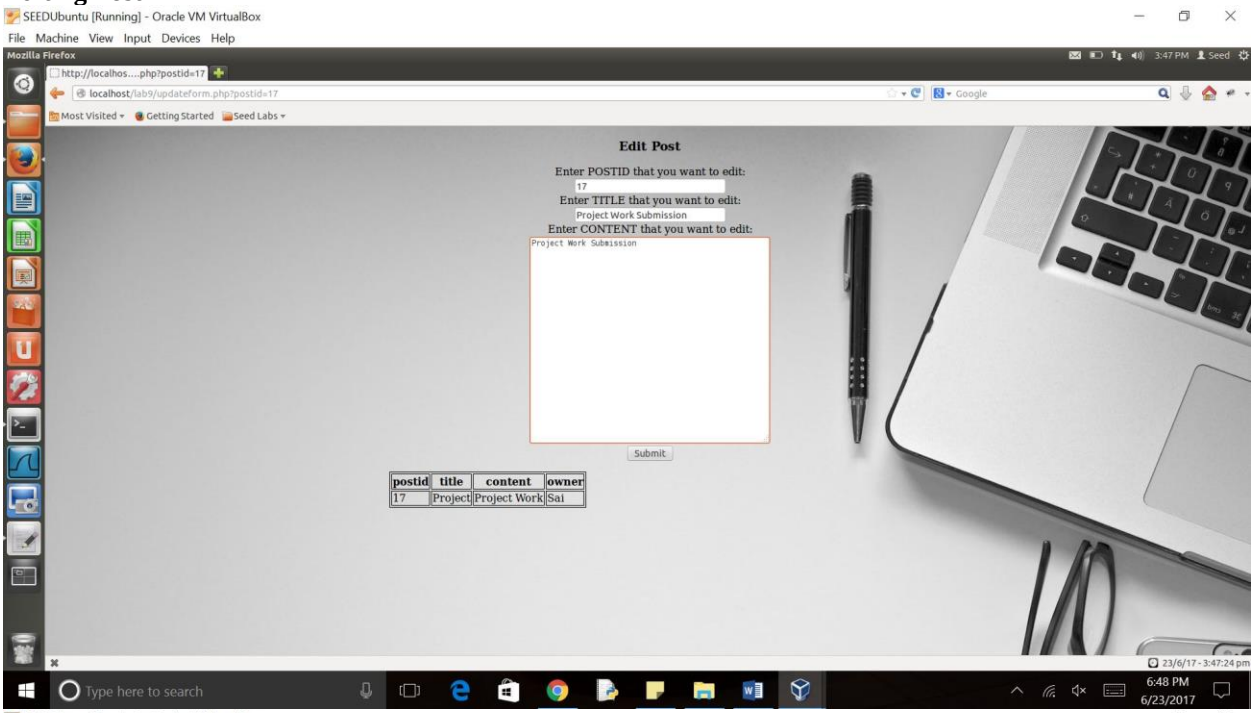


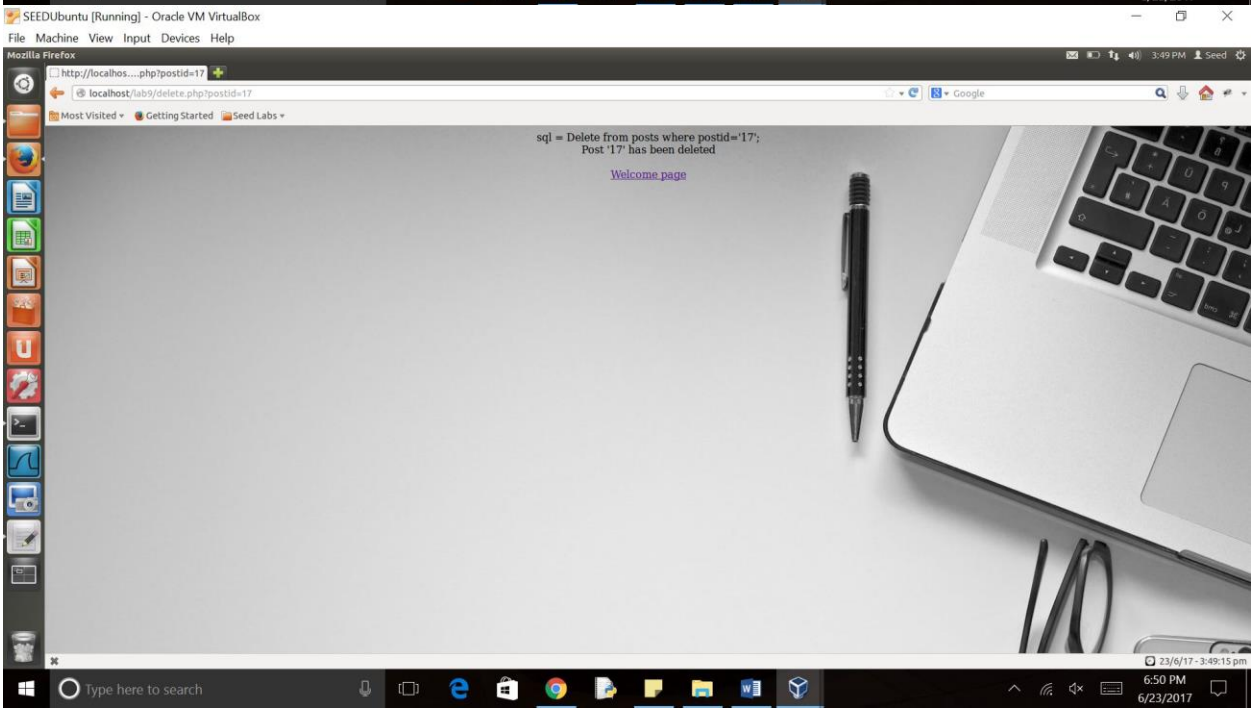
Edit Post


SEEDUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
http://localhos....php?postid=17
localhost/lab9/updateform.php?postid=17
Most Visited Getting Started Seed Labs
Edit Post
Enter POSTID that you want to edit:
Enter TITLE that you want to edit:
Enter CONTENT that you want to edit:
Submit
postid title content owner
17 Project Project Work Sai
Type here to search
6:47 PM
6/23/2017


SEEDUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mozilla Firefox
http://localhost/lab9/edit.php
localhost/lab9/edit.php
Most Visited Getting Started Seed Labs
debug> Postid= 17; Title= Project Work Submission; Content=Project Work Submission
sql = update posts set content ='Project Work Submission',title='Project Work Submission' where postid='17'; post '17' has been updated
Welcome page
Type here to search
6:49 PM
6/23/2017

Editing Post





File Management System

- Authorization:
  auth.php

- About:
  about.php

- Login:
  log.php

- Guest:
  guest.php
  View Comments: guestcomment.php
  Add Comments: guestcform.php

- Welcome page:
  index.php

- Chat:
  websocketchat1.php

- Add User
  Newuserform.php
  Adduser.php

- Change Password
  Changepasswordform.php
  Changepassword.php

- Add Post
  Newpostform.php
  New.php

- Edit Post
  Updateform.php
  Edit.php

- Delete Post
  Delete.php

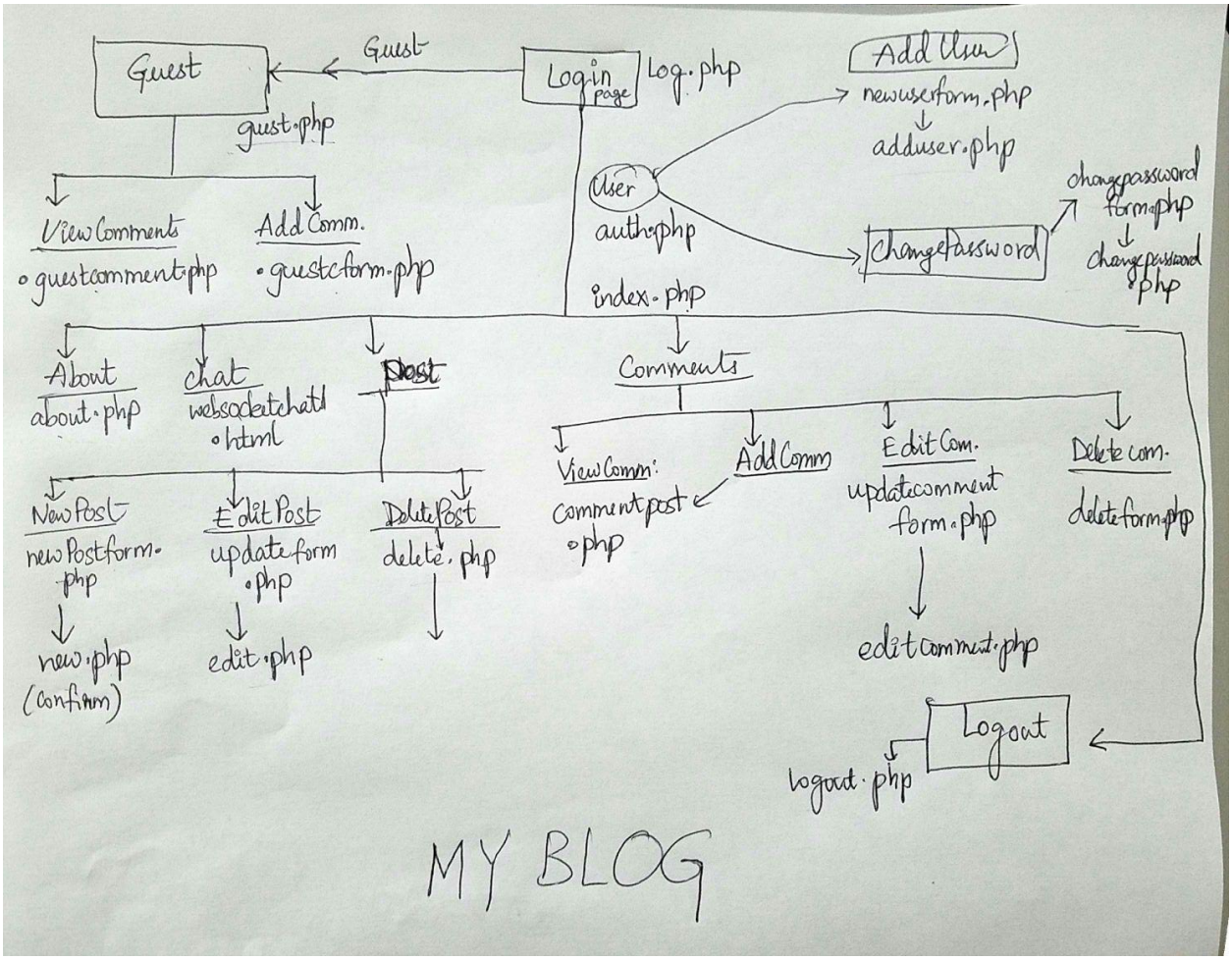
- Add Comment
  Commentpost.php

- Edit Comment

Updatecommentform.php
Editcomment.php

- Delete Comment
  Deleteform.php

- Logout:
  logout.php



File Management System

Potential Risks and preventions adopted:

Common Security risks are: "NEVER TRUST USER INPUT"

1. Session hijacking

Avoid:

a.set session cookies parameters

b. store more information about the browser (HTTP_USER_AGENT, REMOTE_ADDR, HTTP_ACCEPT, HTTP_ACCEPT_LANGUAGE)

c. re-authenticate if there is any change in the database

d. while user request is to change password, ask for current password too

2. SQL Injection Attacks

Avoid:

a. user database platform dependent APIs- eg. mysql_real_escape_string

b. use prepared statements

c. escaping all user supplied input

3. Cross-site Request Forgery

Avoid:

a. server needs to re-authenticate

b. generate secret token to link between the webpages

c. validating referrer

4. Cross-site Scripting Attack

Avoid:

a. use HTML entities

b. use HTML attributes