Source Code

By Ctrl Alt Elite

Christian Laemmerhirt & Jackson Rosenthal

Table of Contents

[**1.1 Project Summary**](#_v0tsjx6ropq1) **3**

[**1.2 Website Overview**](#_g5kc8v80rtht) **3**

[**1.3 UX Wireframes**](#_2vpspt8f53fs) **5**

[1.3.1 Create Post](#_914xcy7fcz0i) 5

[1.3.2 Create Account](#_grrviu3swv7j) 6

[1.3.3 Login](#_tjizogrrv5uk) 7

[1.3.5 Home Page](#_z7l6zxvg9l5e) 7

[**1.4 Description of home.php**](#_qr1u72gx3uqp) **8**

[**1.5 Description of createpost.php**](#_czgiawuqa505) **9**

[**1.6 Database**](#_s2y0szqsw1tb) **9**

[**1.7 User Logins**](#_uresd99s9ab8) **10**

[**1.8 SQL**](#_mcwytznnrx0s) **10**

[**1.9 Testing**](#_bmcxha2nsjax) **13**

[1.9.1 Home Page](#_ikd85ubqmbnm) 13

[1.9.2 Comment on Post](#_6n66wmcwcl2e) 14

[1.9.3 Login](#_w39rzdofuupj) 14

[1.9.4 Create Post](#_7ozlr33bxpqx) 15

[**2.0 User Manual**](#_h5369m85ym5w) **15**

# 1.1 Project Summary

**Team Members:** Christian Laemmerhirt and Jackson Rosenthal

**Team Name:** Ctrl Alt Elite

**Meeting Place:** Collaboration over VoIP services such as Discord, project is shared over home server via SSH and SFTP. Meeting in person will be dedicated only to specific issues that need face to face collaboration.

**Team Leader:** No leader, we will work together and split work evenly depending on what needs to be completed.

**Website Name:** Source Code

**Website Purpose/Description:** Our website’s purpose is for CS students to be able to creates posts about CS related topics with a comment section where others can create comment chains. A point system will be used so people can upvote / downvote each post or comment and have the posts and comments sorted in descending order so the most popular posts and opinions will be at the top of the page. Our website is similar to how Reddit works but specific to CS related topics tailored towards students.

# 

# 1.2 Website Overview

Home Page

The home page is the main page of our website. It consists of menu bar at the top of page taking up the width of the browser. It will have three buttons, ‘Home’, ‘Create Post’, and ‘Login’. Below the menu bar will be a center-aligned section containing all the posts on the website. Each post will consist of the title of the post, how many comments it has, who created the post, how long ago the post was created, as well as how many points the post has. Each post takes up about 104 pixels and continues down the page. Possible feature of limiting the number of posts per page and including different page numbers to the home page. The layout of the website is all HTML/CSS but the data to populate all the posts will be fetched from the database using PHP and SQL. PHP will then be used to loop through all the posts, sorting them by points and time, generating the HTML to layout the page.

Create Post

Clicking the ‘Create Post’ button will redirect the user, if logged in, to a page where they can create a post with a specified title and body content. Below the text boxes is a ‘Create Post’ button which will send the post to the database using PHP and SQL which will then add it to the home page.

Login/Create Account

Clicking the ‘Login’ button will keep the user on the home page but bring up a small window in the center of the browser allowing the user to enter their username and password. They will be able to click ‘Remember Me’ in order to keep the user logged in. If the user does not have an account, a ‘Create Account’ button will be located under the text fields which will change the login window to a create account window. Here, the user can enter a desired username and password with a password confirmation textbox as well. This window will be created using purely HTML/CSS with using PHP and SQL to verify and store usernames and passwords for logging in or creating accounts.

# 

# 

# 1.3 UX Wireframes

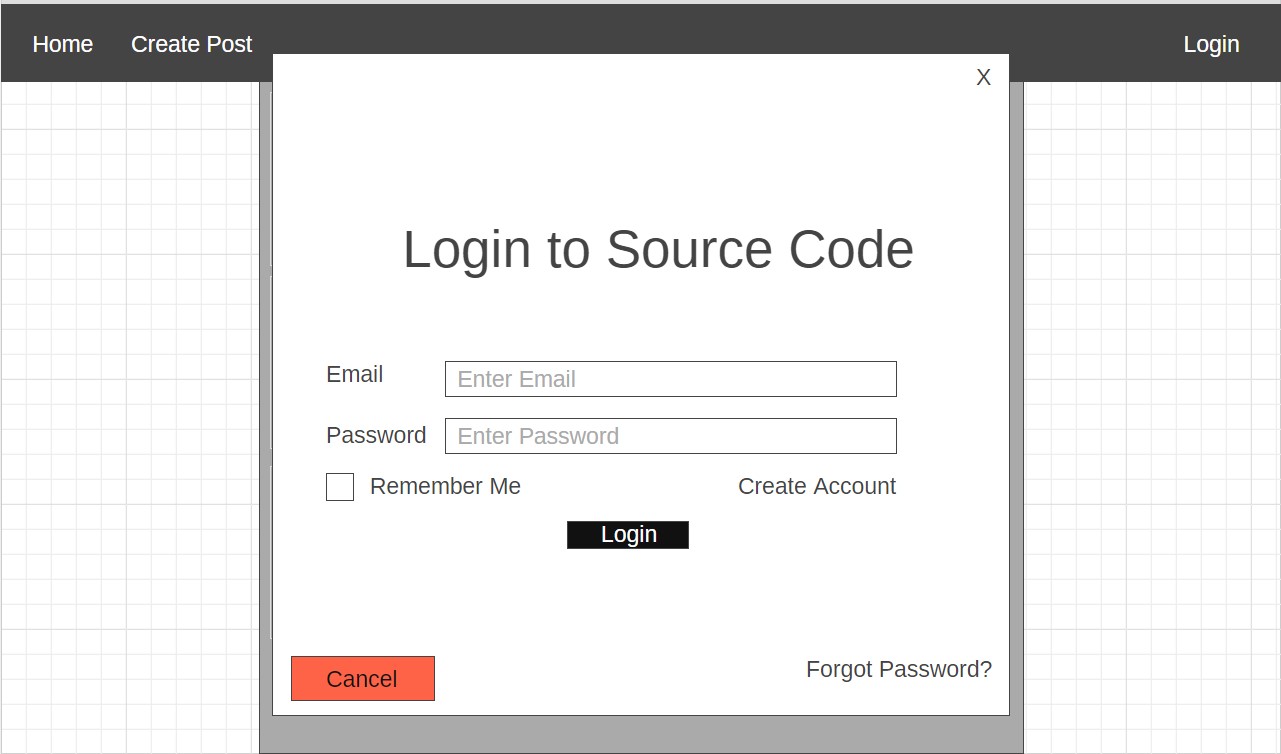
## 1.3.1 Create Post

## 

## 1.3.2 Create Account

## 1.3.3 Login

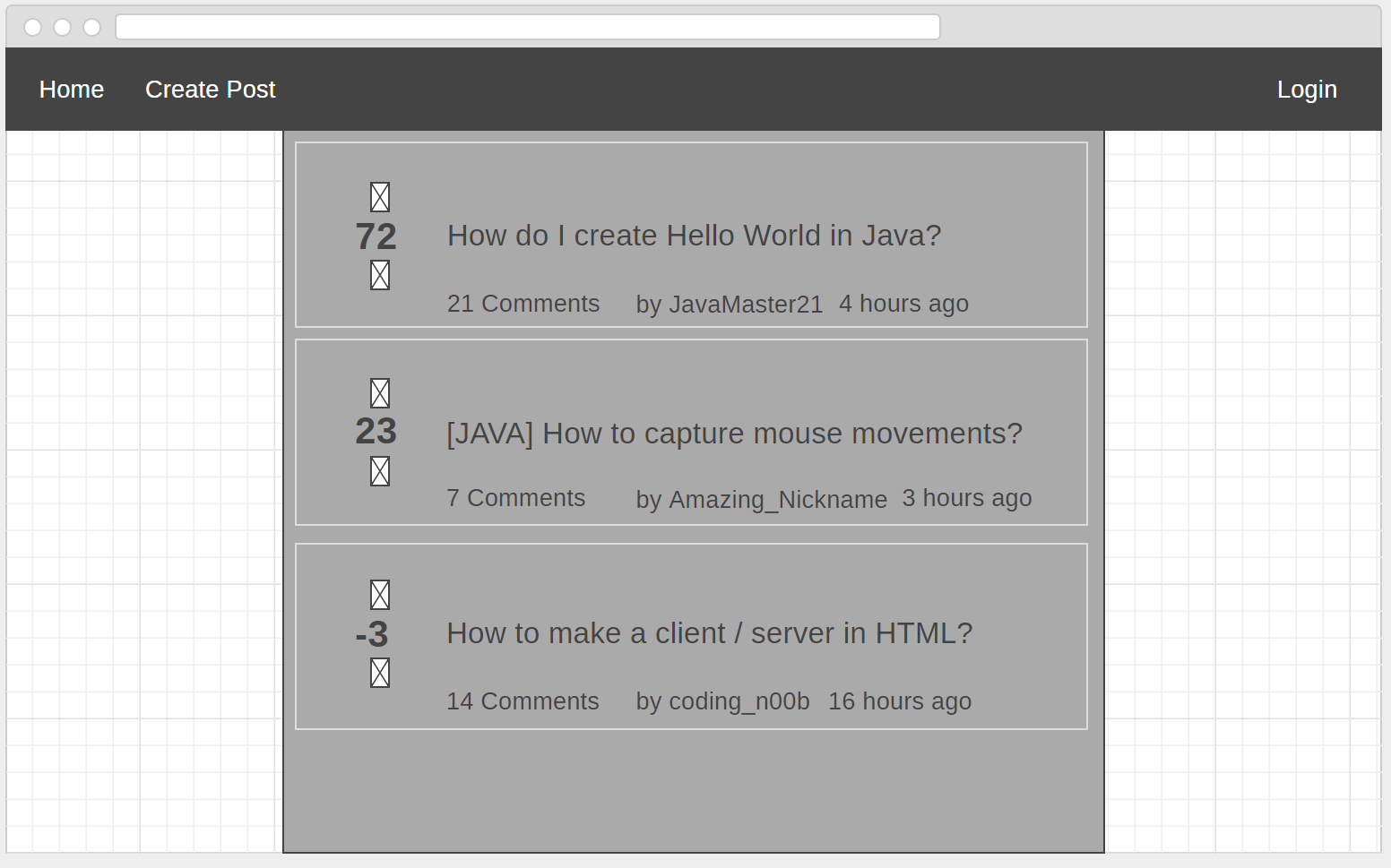
## 



## 

## 

## 1.3.5 Home Page



## 

## 

## 

## 

## 

## 

## 

## 

## 

# 1.4 Description of home.php

**Database Connection**

To connect to the database, a PDO object was created using:

Host: localhost:22556

Database Name: 337Website

Username: csuser

Password: csit337

Query statements were prepared using PDO->prepare() and then executed. We then fetched the table data using $statement->fetchAll(PDO::FETCH\_ASSOC).

**Pre-fetch data**

The home page displays all the posts on the server as well as being able to process logins and creating an account. Therefore, a php script was used to fetch all the posts and their contents and store them in an array.

**Generating posts**

Another script was used to fill in all the post data into an HTML template. HTML and CSS were used to create the structure of the post and then a loop was used to iterate over all the posts in the array, populating the HTML with data.

**Login**

In order to login, A SELECT statement was used to fetch the entered username. If the username does not exist in the database, the user is prompted with ‘Invalid Username or Password’. If the username does exist, the password hash is fetched and compared to the entered password. If the username and password match the record on the database, the user is redirected to the homepage with the ability to comment and make posts.

**Create Account**

The script for creating an account works similarly. First, a SELECT statement is used to fetch the entered username. If a record is returned, the user is prompted with ‘Username Taken’. If not, a new record is added to database with the next ID in the table, the entered username, and a hash of the entered password. The user is then redirected to the homepage and is able to create posts.

Login sessions are stored in the $\_SESSION array.

# 1.5 Description of createpost.php

**Submit Post**

A script was used to check whether the submit button had been pressed. If pressed, a new posts record is created in the database with the the next post ID, post title, content, user, and date. Then the page is refreshed to show the new comment.

# 1.6 Database

**Database**

* Posts
  + ID
  + Title
  + Content
  + CreatedByUserID
  + Points
  + TimePosted (UNIX Epoch Time)
* Comments
  + PostID
  + Content
  + CreatedByUserID
  + Points
  + TimePosted (UNIX Epoch Time)
* Users
  + ID
  + Username
  + PasswordHash

# 1.7 User Logins

Username: **codemaster22**

Password: **password**

Username: **n00bcoder**

Password: **password**

Username: **1337h4xboii**

Password: **password**

Username: **admin**

Password: **password**

# 1.8 SQL

CREATE DATABASE `337Website`;

CREATE TABLE `Comments` (

`PostID` int(11),

`CommentID` int(11),

`Content` varchar(2000),

`CreatedByUserID` int(11),

`Points` int(11),

`TimePosted` timestamp DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO `Comments` (`PostID`, `CommentID`, `Content`, `CreatedByUserID`, `Points`, `TimePosted`) VALUES

(1, 1, 'Great post!', 2, 2, '2018-11-27 20:38:45'),

(2, 1, 'A class where you can have some methods instantiated. Like an interface with some implemented methods.', 1, 3, '2018-11-28 00:26:27'),

(2, 2, 'Like an interface with some concrete methods.', 3, 1, '2018-12-04 19:35:54'),

(5, 2, 'RSA uses a public key to encrypt data and a private key to decrypt. You use the other person\'s public key to encrypt something and send it to them. Then they use their private key to decrypt it.', 3, 1, '2018-12-04 20:30:30'),

(3, 2, 'HTML', 1, 2, '2018-12-04 20:31:00'),

(1, 2, 'section .text\r\nglobal \_start ;must be declared for linker (ld)\r\n\r\n\_start: ;tell linker entry point\r\n\r\n mov edx,len ;message length\r\n mov ecx,msg ;message to write\r\n mov ebx,1 ;file descriptor (stdout)\r\n mov eax,4 ;system call number (sys\_write)\r\n int 0x80 ;call kernel\r\n\r\n mov eax,1 ;system call number (sys\_exit)\r\n int 0x80 ;call kernel\r\n\r\nsection .data\r\n\r\nmsg db \'Hello, world!\',0xa ;our dear string\r\nlen equ $ - msg ;length of our dear string\r\n\r\nCopied straight from google', 1, 3, '2018-12-04 20:41:50'),

(7, 2, 'Use a PDO object to connect. Write a query statement, prepare it and execute it.\r\nHere\'s an example:\r\n\r\n$dbname = \"mysql:host=localhost;port=22556;dbname=337Website\";\r\n$dbuser = \"csuser\";\r\n$dbpass = \"csit337\";\r\n\r\n\r\n$conn = new Pdo($dbname, $dbuser, $dbpass, array(PDO::ATTR\_PERSISTENT => false) );\r\n$query = \"SELECT \* FROM Posts ORDER BY Points DESC\";\r\n$statement = $conn->prepare($query);\r\n$statement->execute();\r\n\r\n$posts = $statement->fetchAll(PDO::FETCH\_ASSOC);', 1, 1, '2018-12-04 20:43:31'),

(3, 3, 'JavaScript', 2, 1, '2018-12-04 21:17:38'),

(6, 2, 'Use Socket(port) and ServerSocket(ip, port)', 2, 2, '2018-12-06 15:38:56'),

(5, 3, 'AES uses a single key to encrypt and decrypt stuff.\r\n\r\nRSA uses two keys a public and private key', 1, 1, '2018-12-10 18:34:51'),

(6, 3, 'Google it', 1, 1, '2018-12-13 18:09:47');

CREATE TABLE `Posts` (

`ID` int(11),

`Title` varchar(175),

`Content` varchar(2000),

`CreatedByUserID` int(11),

`Points` int(11),

`TimePosted` timestamp DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO `Posts` (`ID`, `Title`, `Content`, `CreatedByUserID`, `Points`, `TimePosted`) VALUES

(1, 'How do I print hello world in assembly?', 'How would I go about display hello world in a console in assembly? Thanks.', 1, 18, '2018-11-30 21:00:33'),

(2, 'What is an abstract class?', 'In Java, what\'s an abstract class? What\'s it used for and why not just use an interface or regular class or something. Java is too confusing I\'ll never be good at coding.', 2, 5, '2018-12-03 17:59:13'),

(3, 'What is the best programming language?', 'Out of all the programming languages which is the best of the best. Which language should I use for everything and never touch another language again? I wanna become the worlds best coder.', 2, -3, '2018-12-03 17:59:16'),

(5, 'How does RSA encryption work', 'I\'ve heard of AES encryption which uses one key to encrypt and decrypt data. but how does RSA work with multiple keys?', 2, 1, '2018-11-30 21:00:47'),

(6, 'How do I make a client / sever in Java?', 'I want to make a program where I can run a server and connect to it remotely over the internet and send data.', 3, 2, '2018-12-04 20:16:30'),

(7, 'How to use PHP to get data from a database?', 'I have a website I want to use PHP on in order to fill in fields with data from a database.', 3, 1, '2018-12-04 20:28:27');

CREATE TABLE `Users` (

`ID` int(11),

`Username` varchar(30),

`PasswordHash` varchar(500)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `Users` (`ID`, `Username`, `PasswordHash`) VALUES

(1, 'codemaster22', '$2y$10$I0dPkoux8biTLe3jjMf7KeKuE30MBQ.G/EMmRialZVJWqyaeAg9ei'),

(2, 'n00bcoder', '$2y$10$6JOjKqAjYpX.g.rW2pkqt.zE.AS9ndpYIaolnVfSwUi0pTfaJWPrK'),

(3, '1337h4xboii', '$2y$10$fzafNJj85Dhxx/FNPWczHe0HCqvfhrJ9Fx9aMoHDYfknEOaQfVcv6'),

(4, 'admin', '$2y$10$WSf5F8gTgqwkXsNi1xXB/ufbSUGVFIohyXLZk1IrYhqrI/RIY41Y6');

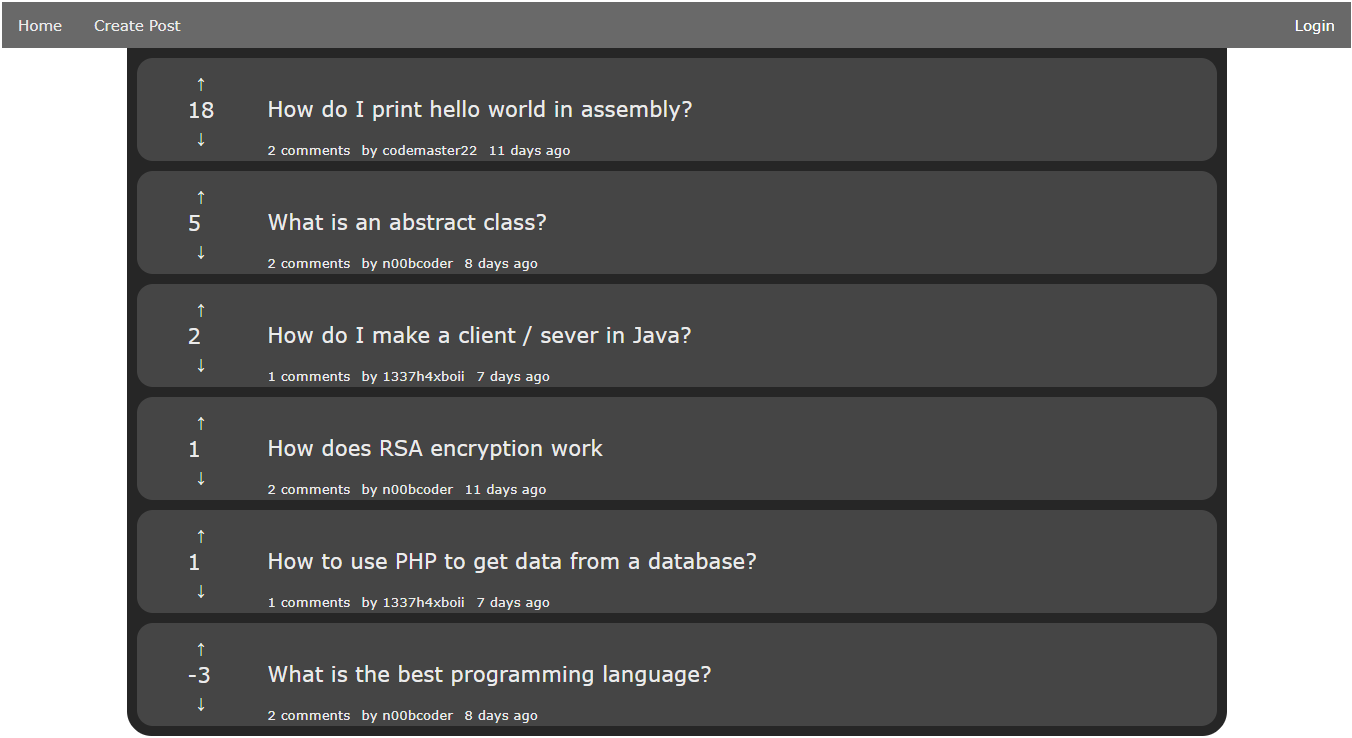
;

CREATE USER 'csuser@localhost' IDENTIFIED BY 'csit337';

GRANT ALL PRIVILEGES ON 337Website TO 'csuser@localhost' WITH GRANT OPTION;

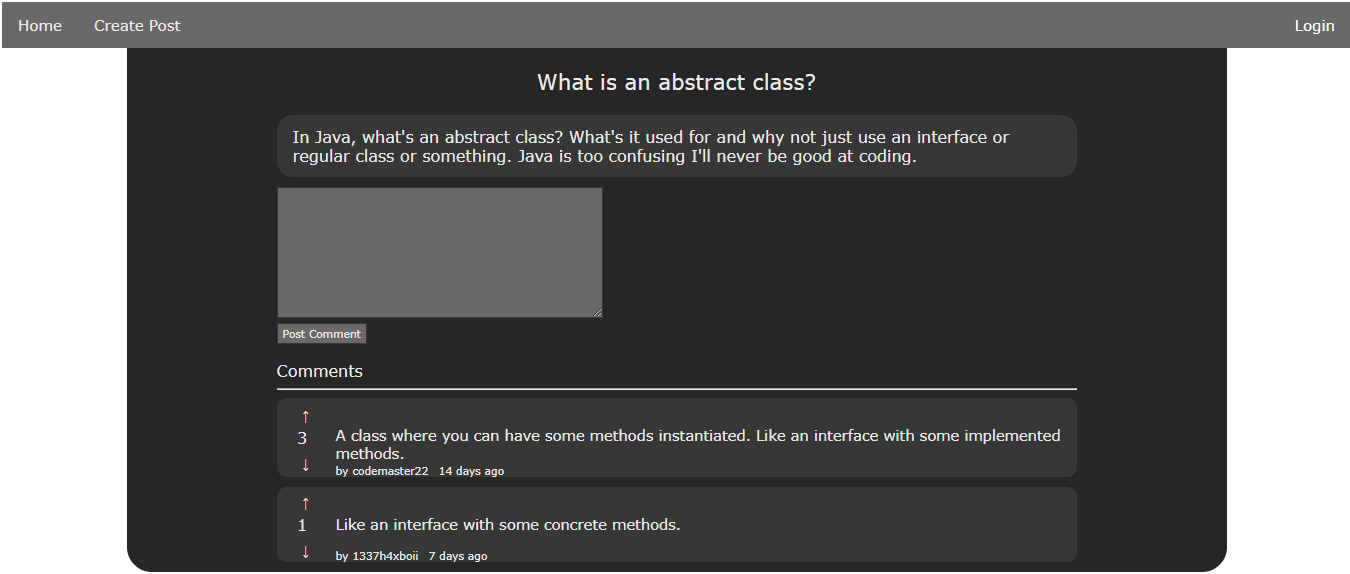
# 1.9 Testing

## 1.9.1 Home Page



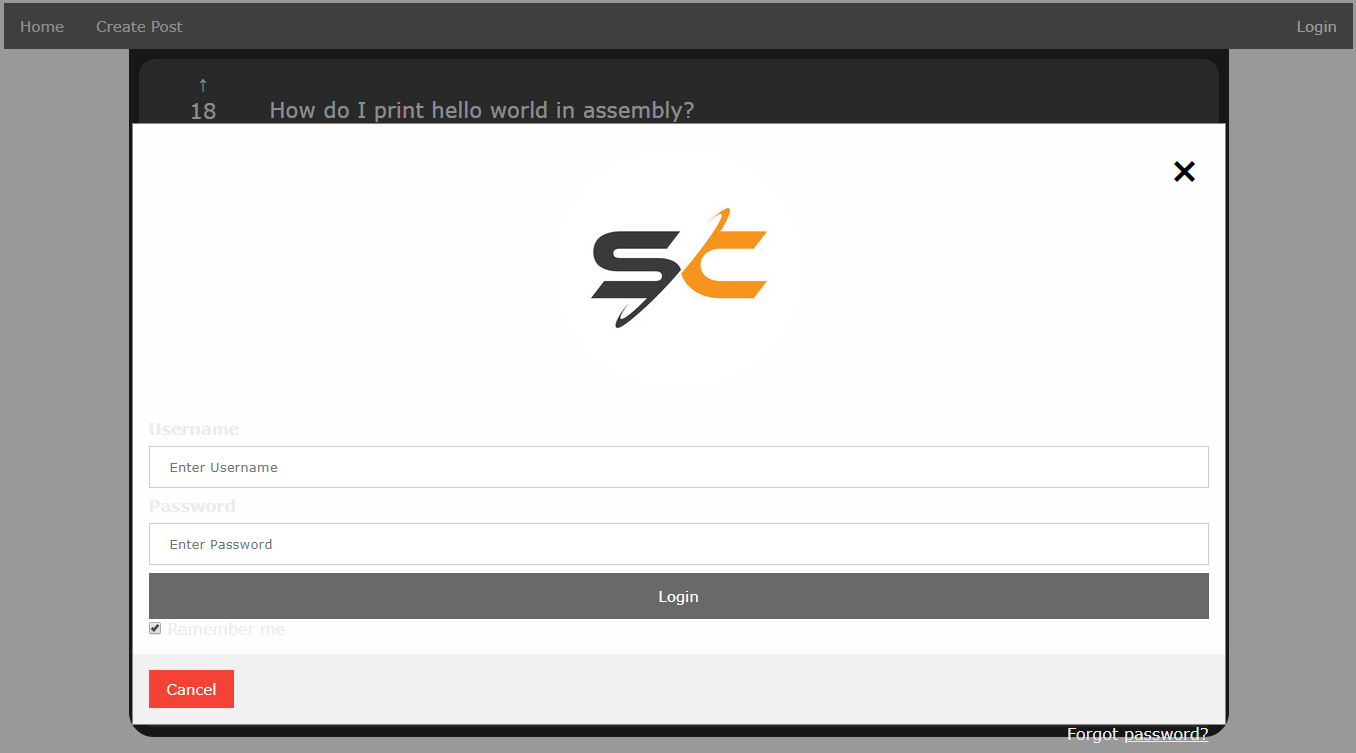
Once you have connected to the site you are greeted by the home page. This page lists all the posts by the users on the website in order by most popular, this is determined by the “upvotes” or “downvotes”. You can create a post if you are logged in, you can return to the home page, login, and view the posts. The home page displays information about each post: how many comments it has, who created the post, how long ago it was created, the title of the post and how many upvotes/downvotes it has.

## 1.9.2 Comment on Post



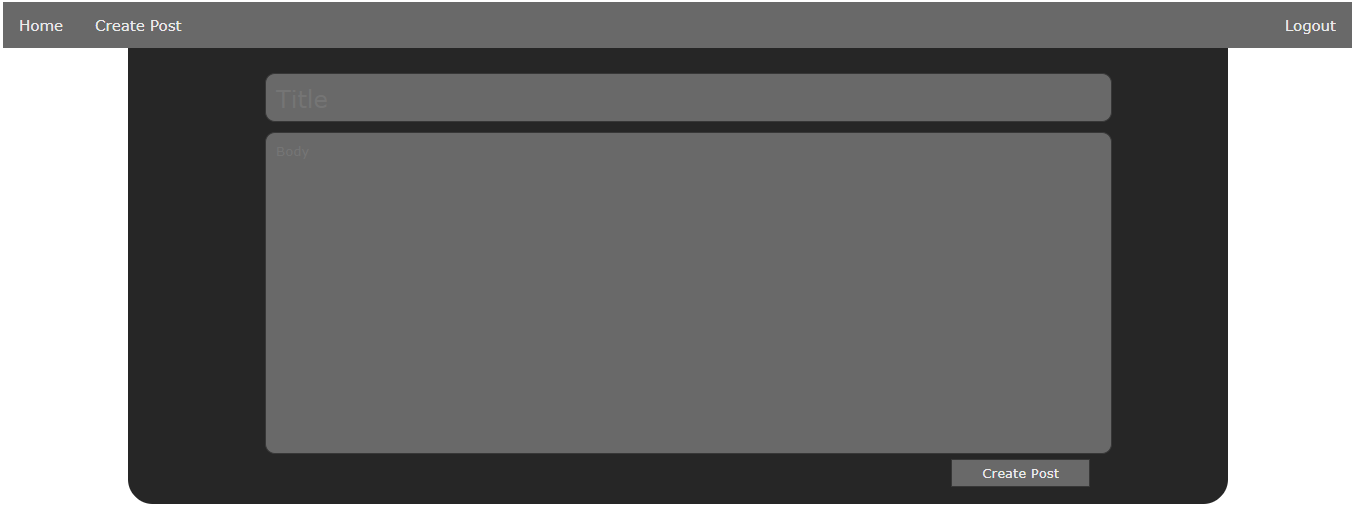
Once you have logged in you will be able to comment on a post. By clicking on the post on the homepage you will be able to view more information about the post. This includes; the content of the post, the comments under the post, the upvotes/downvotes for each comment, who posted the comment and how long ago it was posted.

## 1.9.3 Login



When you click the login button on the homepage you will be greeted with this pop up window. This window will allow you to login. Simply type the username and password and click login.

## 1.9.4 Create Post



After you have logged in you are able to create posts. You have to add a title to the post and then type in the contents of the post. Once you have finished you click create post and you are finished.

## 

# 2.0 User Manual

Login

Click the “Login” button in the top right corner of the screen. A popup window will appear and ask for your username and password. Enter your credentials and click “Login”. The page should refresh and replace “Login” with “Logout”

Logout

Simply click “Logout in the top right corner of the screen. If “Logout” is not there, you are not logged in.

Create Post

First, you must be logged in. Click “Create Post”. You will be prompted to enter a title and body. To post click the “Create Post” button.

Comment

First, you must be logged in. Click on the post you want to comment on. Enter your comment in the provided textbox and click “Post Comment”.

Upvote / Downvote

First, you must be logged in. Click the up arrow to upvote and the down arrow to downvote any post or comment.

Admin Manage User

Logging in with the username, “admin” unlocks the Manage Users tab on the home page. Here just click the button labeled “Change Password” or “Delete User” to change their password or delete the user.

Deleting Posts

Simply click “delete” in the post or comment details. Only the admin to delete posts and comment that he/she didn’t create.