

AIM

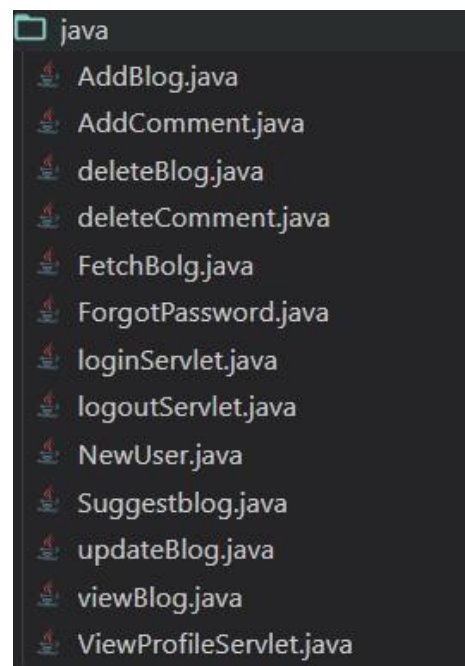
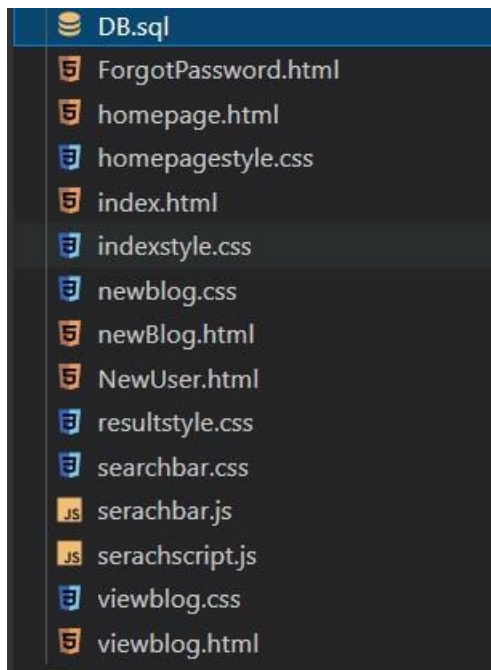
To create a blog creating system

Hardware and Software Requirements

Java, html, javaScript, ccs

Netbeans, apache tomcat, chrome browser

Code



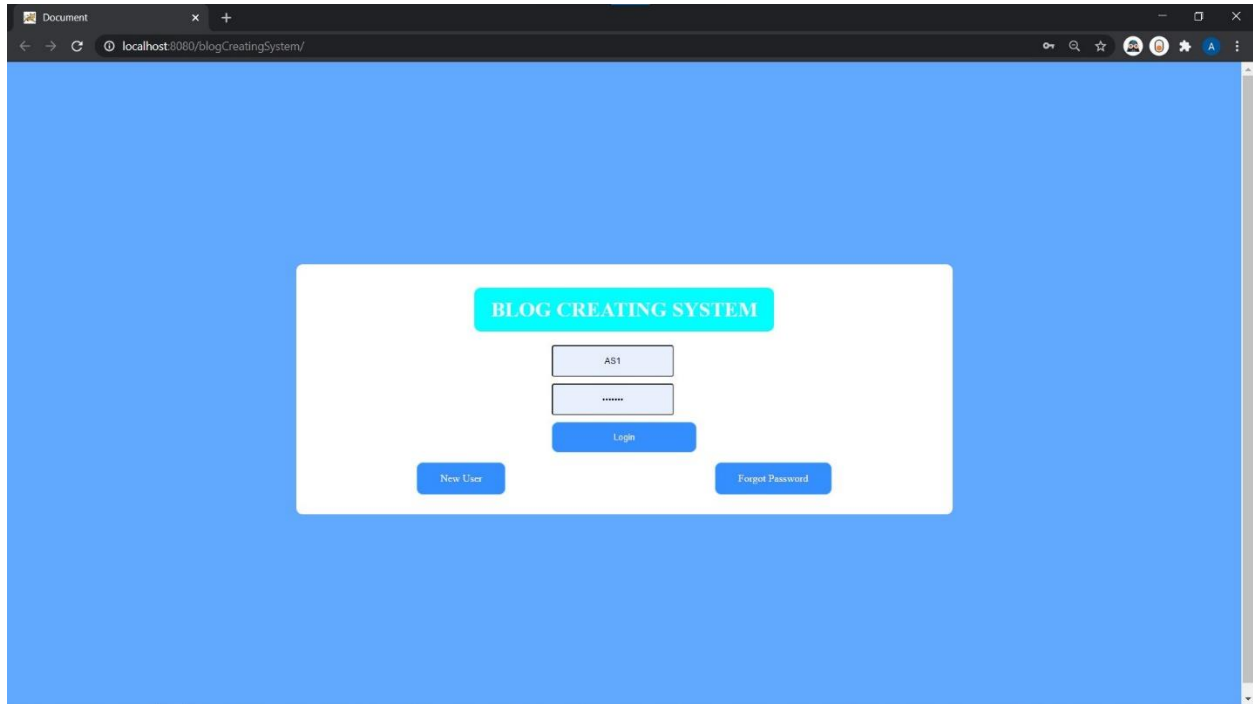
Database

```
drop table image;
drop table tags;
drop table comment;
drop table blog;
drop table user;
create table user (
    uid VARCHAR(10) PRIMARY KEY,
    pwd VARCHAR(15) NOT NULL,
    email VARCHAR(50) NOT NULL,
    name VARCHAR(25) NOT NULL,
    bio VARCHAR(500),
    dob date NOT NULL
);
```

```
create table blog (  
    bid VARCHAR(30) PRIMARY KEY,  
    uid VARCHAR(10),  
    bname VARCHAR(15) NOT NULL,  
    bdate DATE NOT NULL,  
    likes int,  
    dislikes int,  
    content VARCHAR(5000),  
    FOREIGN KEY (uid) REFERENCES USER(uid)  
);  
create table image (  
    blog_id VARCHAR(30),  
    img_id int AUTO_INCREMENT PRIMARY KEY,  
    img VARCHAR(500) NOT NULL,  
    FOREIGN KEY(blog_id) REFERENCES blog(bid) ON DELETE CASCADE  
    )ENGINE=INNODB;  
  
CREATE TABLE TAGS(  
    blog_id VARCHAR(30) NOT NULL,  
    tag VARCHAR(20) NOT NULL,  
    FOREIGN KEY(blog_id) REFERENCES blog(bid) ON DELETE CASCADE  
);  
create table comment (  
    comment_id int PRIMARY KEY AUTO_INCREMENT,  
    uid VARCHAR(10) NOT NULL,  
    bid VARCHAR(30) NOT NULL,  
    content VARCHAR(500) NOT NULL,  
    likes int NOT NULL,  
    dislikes int NOT NULL,  
    cdate date,  
    FOREIGN KEY (uid) REFERENCES USER(uid),  
    FOREIGN KEY (bid) REFERENCES blog(bid)  
);
```

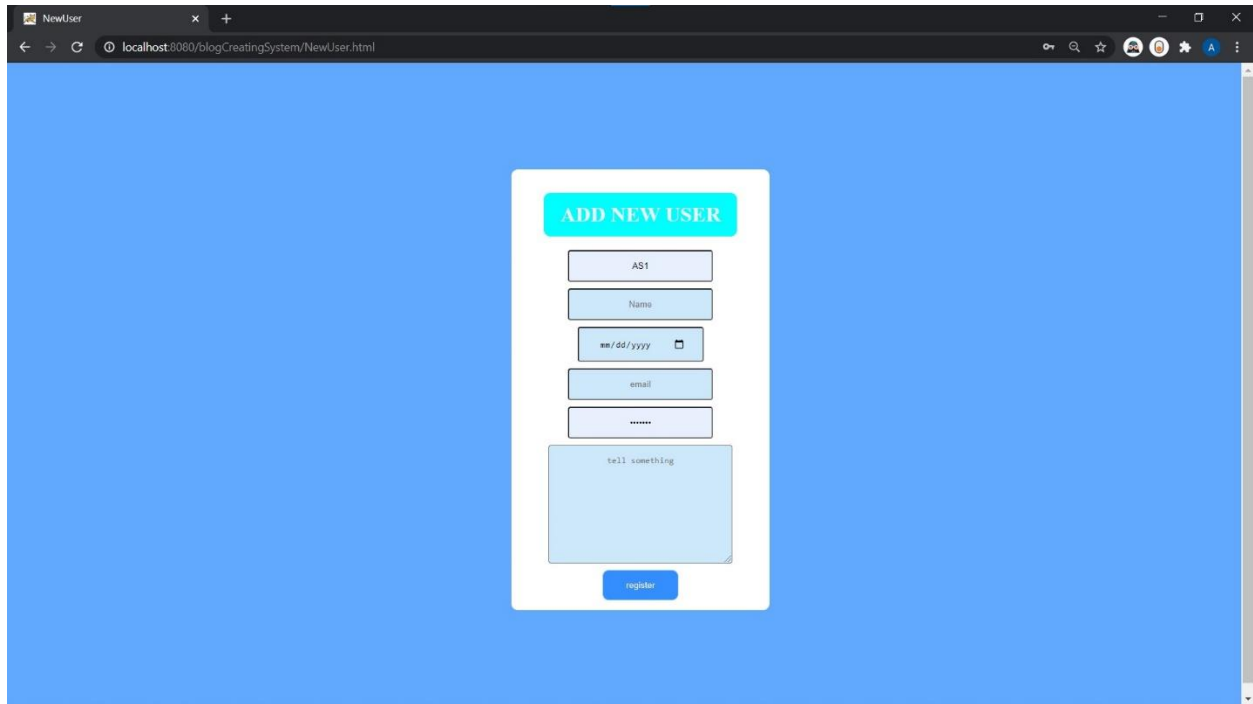
Output:

Login



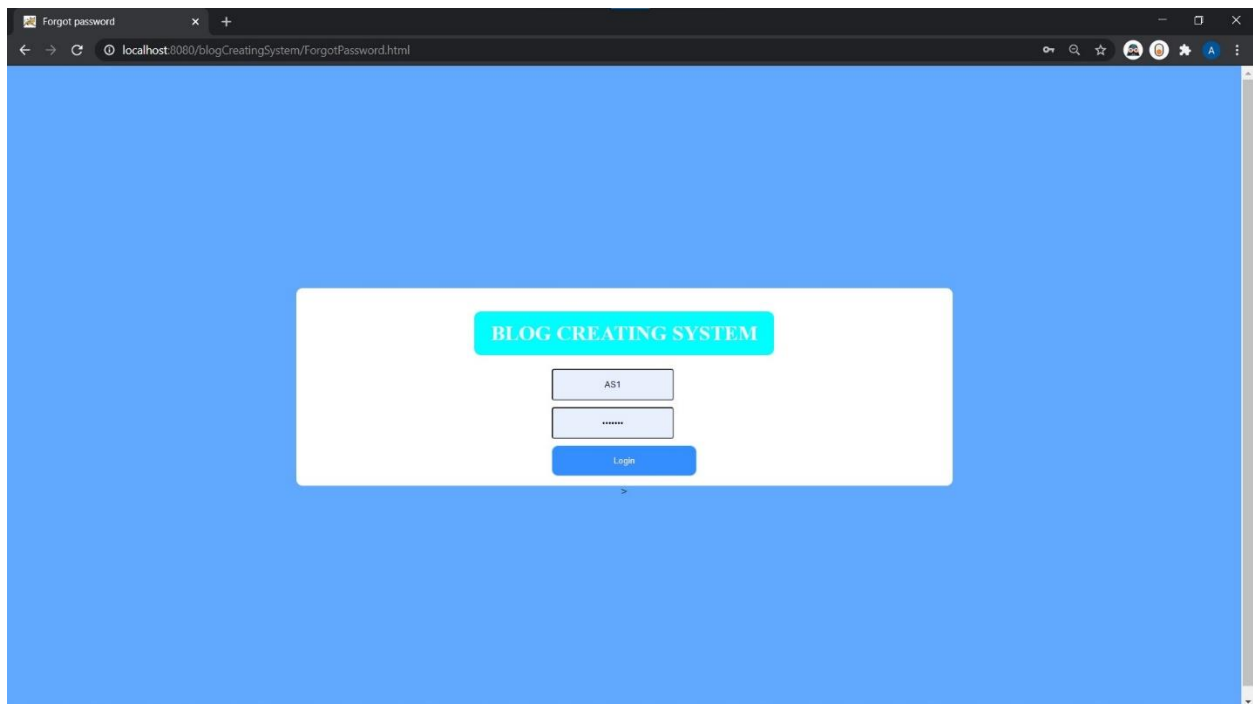
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/blogCreatingSystem/'. The page has a solid blue background. In the center, there is a white rectangular box containing the login form. At the top of this box is a red button labeled 'BLOG CREATING SYSTEM'. Below it are two input fields: the first contains the text 'AS1' and the second contains a series of dots representing a password. A red 'Login' button is positioned below the password field. At the bottom of the white box, there are two red buttons: 'New User' on the left and 'Forgot Password' on the right.

New user



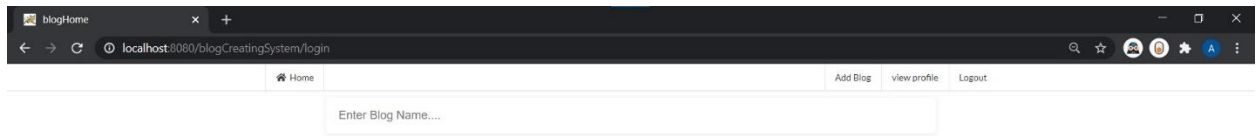
The screenshot shows a web browser window with the title 'NewUser' and the URL 'localhost:8080/blogCreatingSystem/NewUser.html'. The page has a solid blue background. In the center, there is a white rectangular form. At the top of the form is a red button with the text 'ADD NEW USER'. Below this button are several input fields: a text field containing 'AS1', a text field labeled 'Name', a date field with a placeholder 'mm/dd/yyyy' and a calendar icon, a text field labeled 'email', a password field with a placeholder '*****', and a larger text area with a placeholder 'tell something'. At the bottom of the form is a red button labeled 'register'.

Forgot password

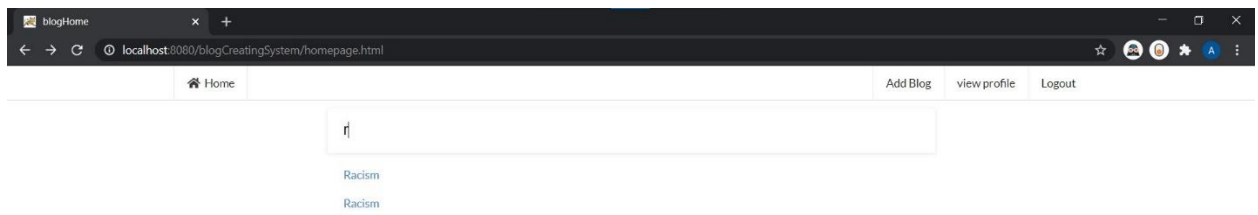


The screenshot shows a web browser window with the title 'Forgot password' and the URL 'localhost:8080/blogCreatingSystem/ForgotPassword.html'. The page has a solid blue background. In the center, there is a white rectangular form. At the top of the form is a red button with the text 'BLOG CREATING SYSTEM'. Below this button are two input fields: a text field containing 'AS1' and a password field with a placeholder '*****'. At the bottom of the form is a red button labeled 'Login'.

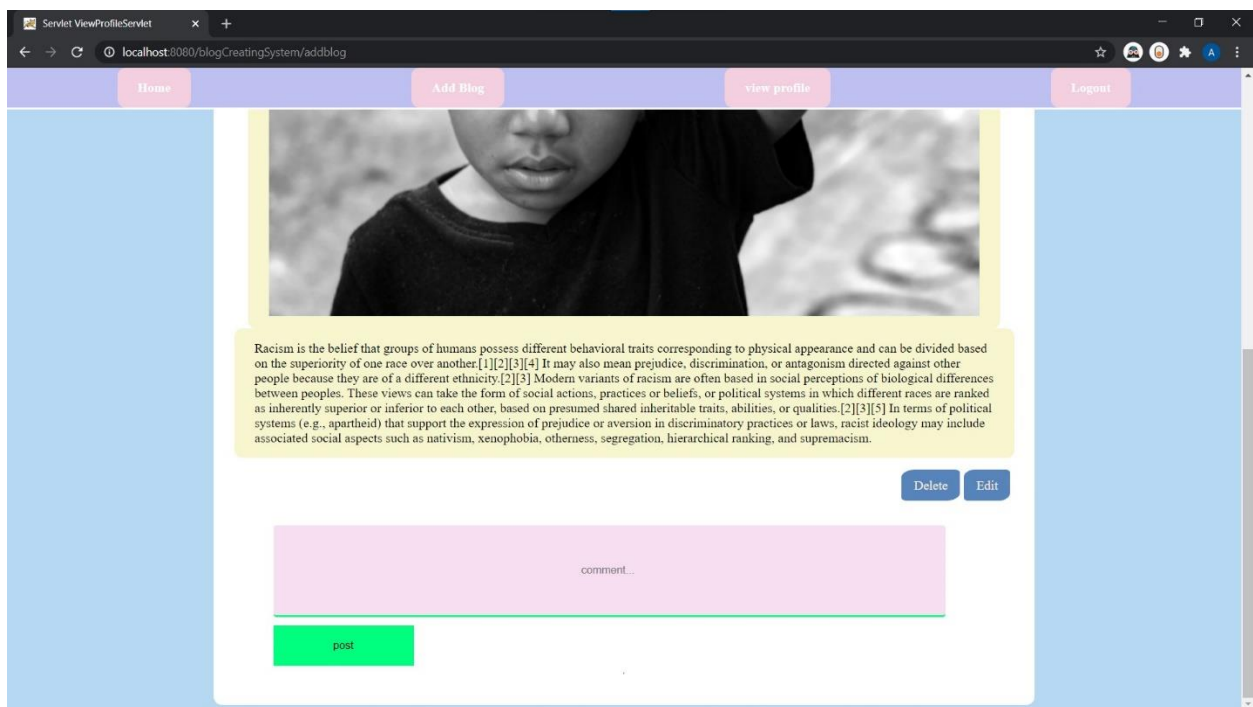
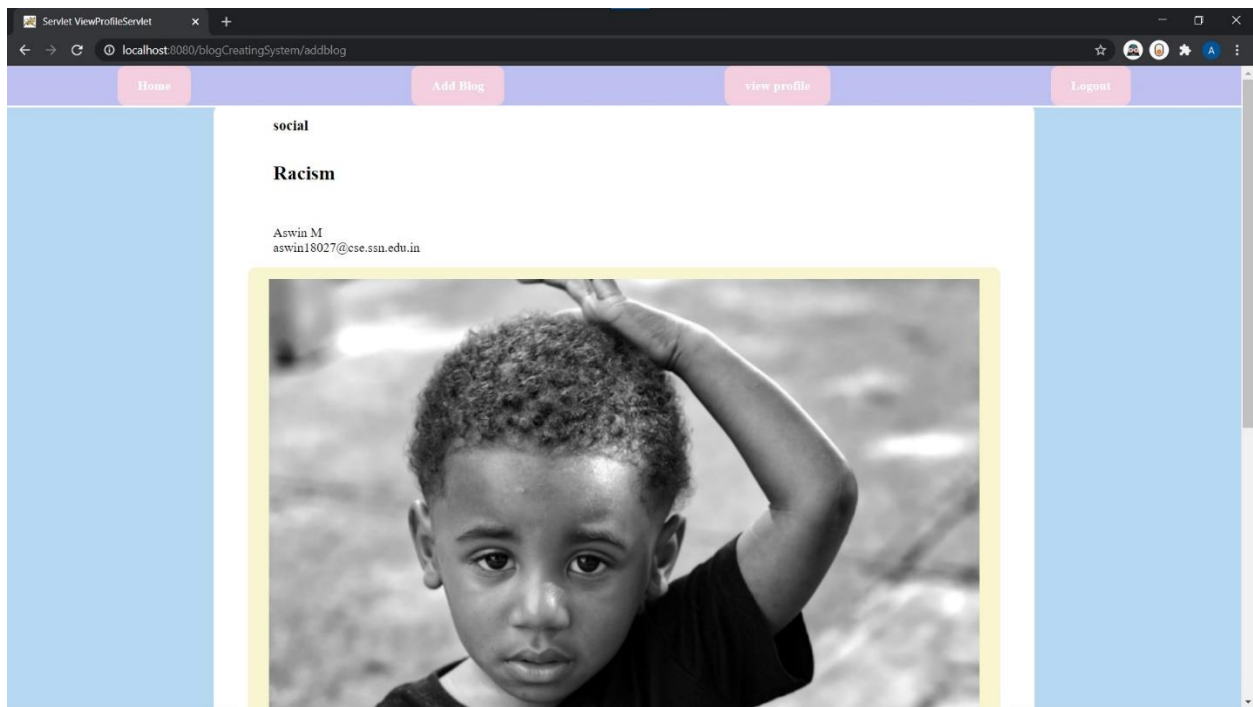
Home page



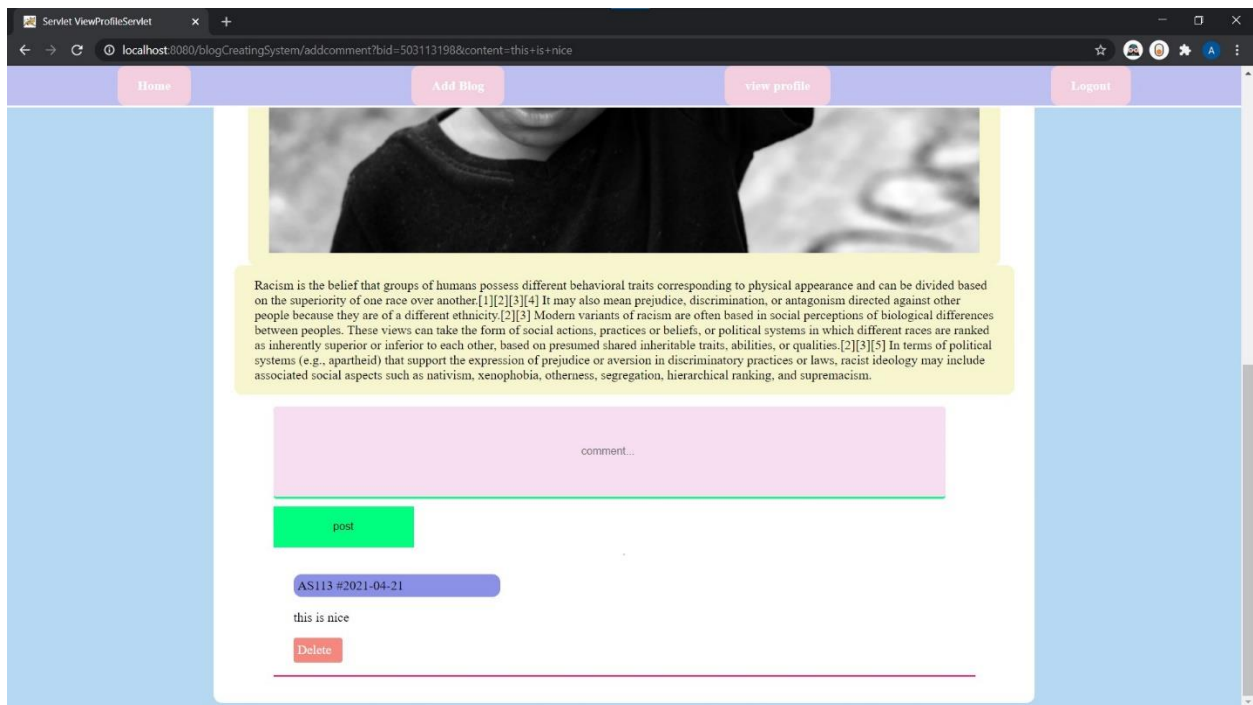
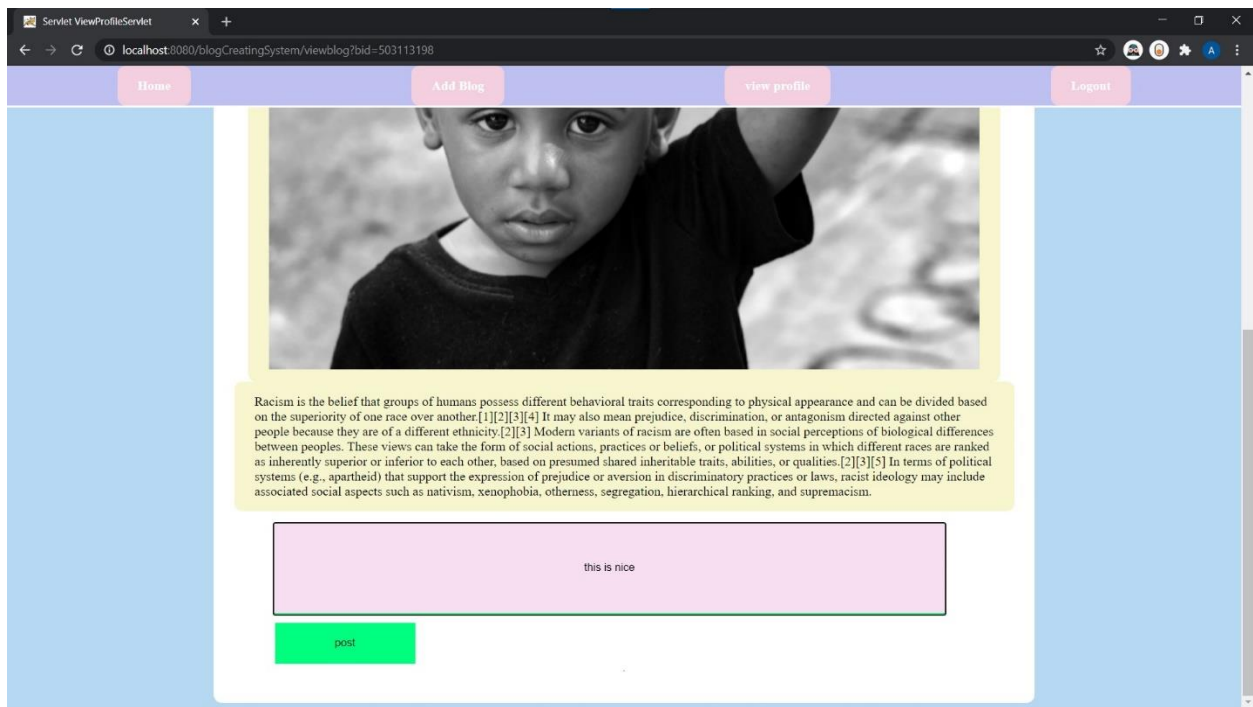
Search blog



View blog

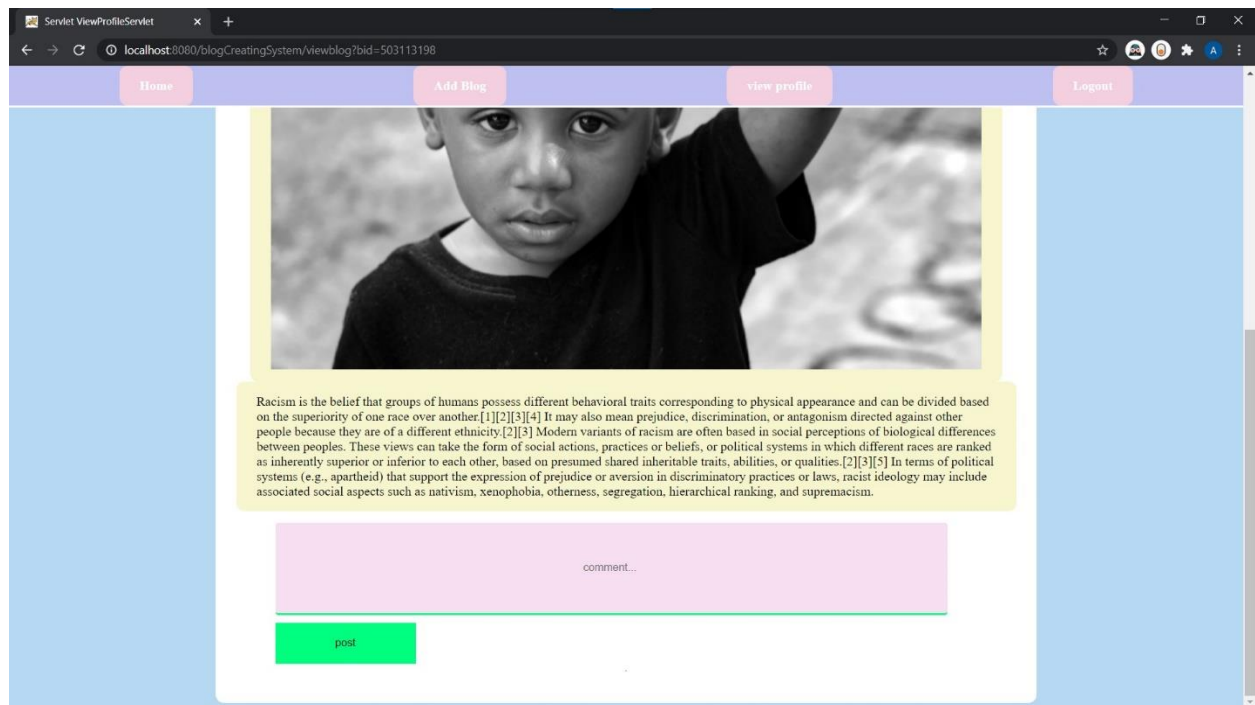


Add comment

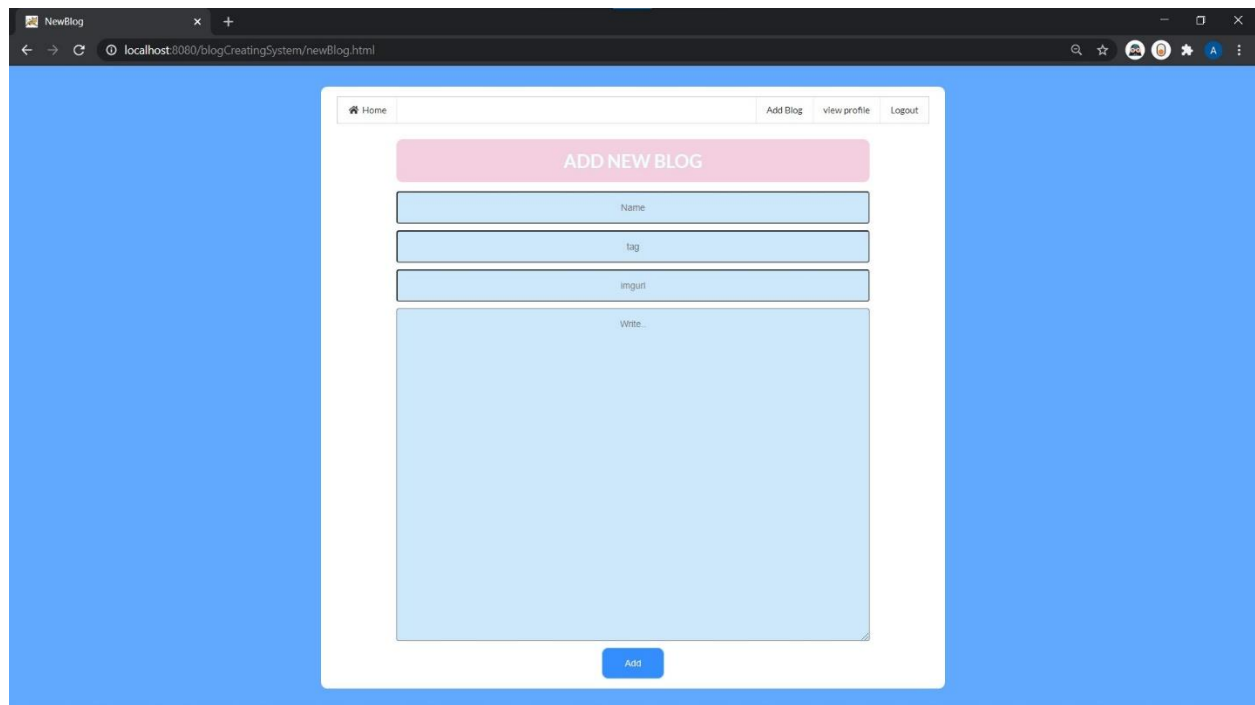


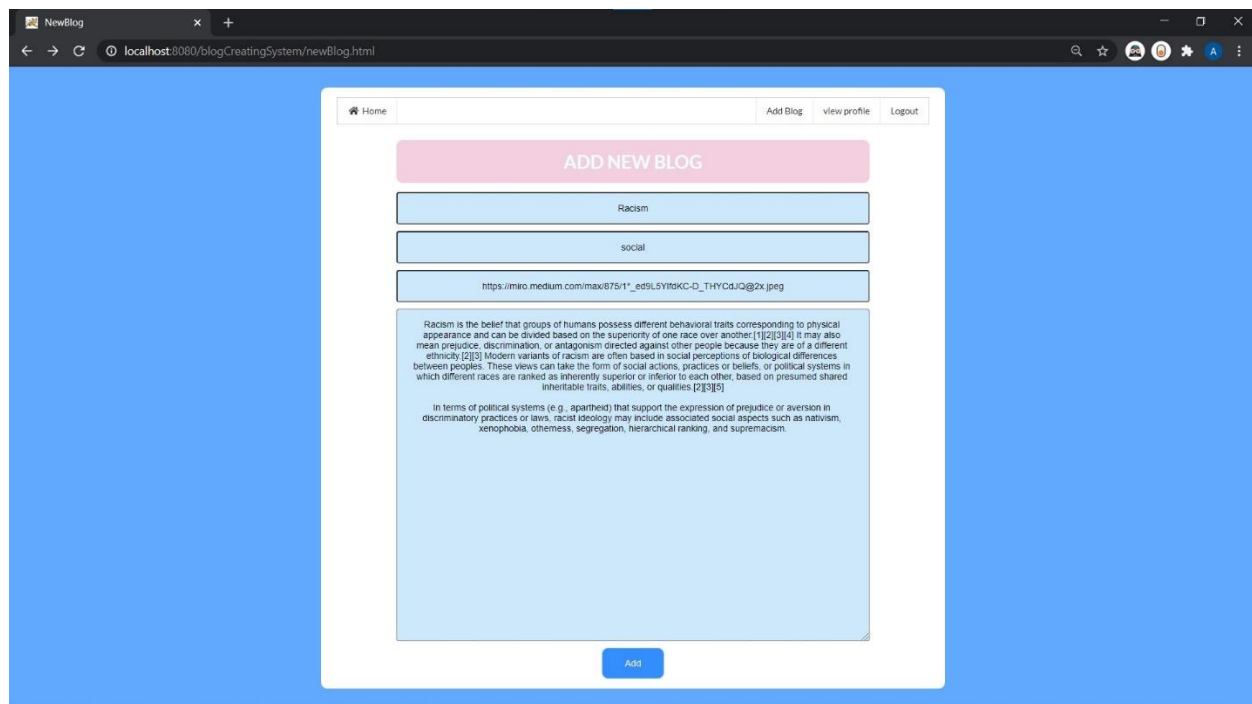
View blog as different user

(the person who created can only edit or delete the blog)

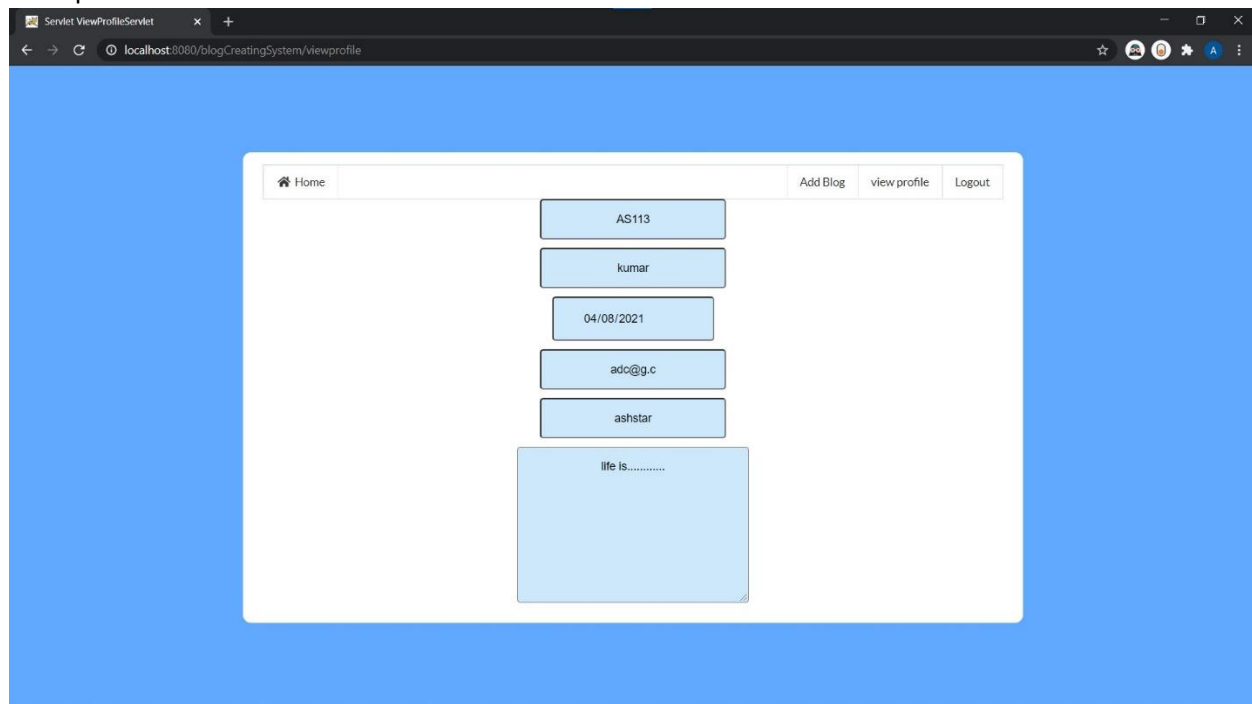


Add new blog

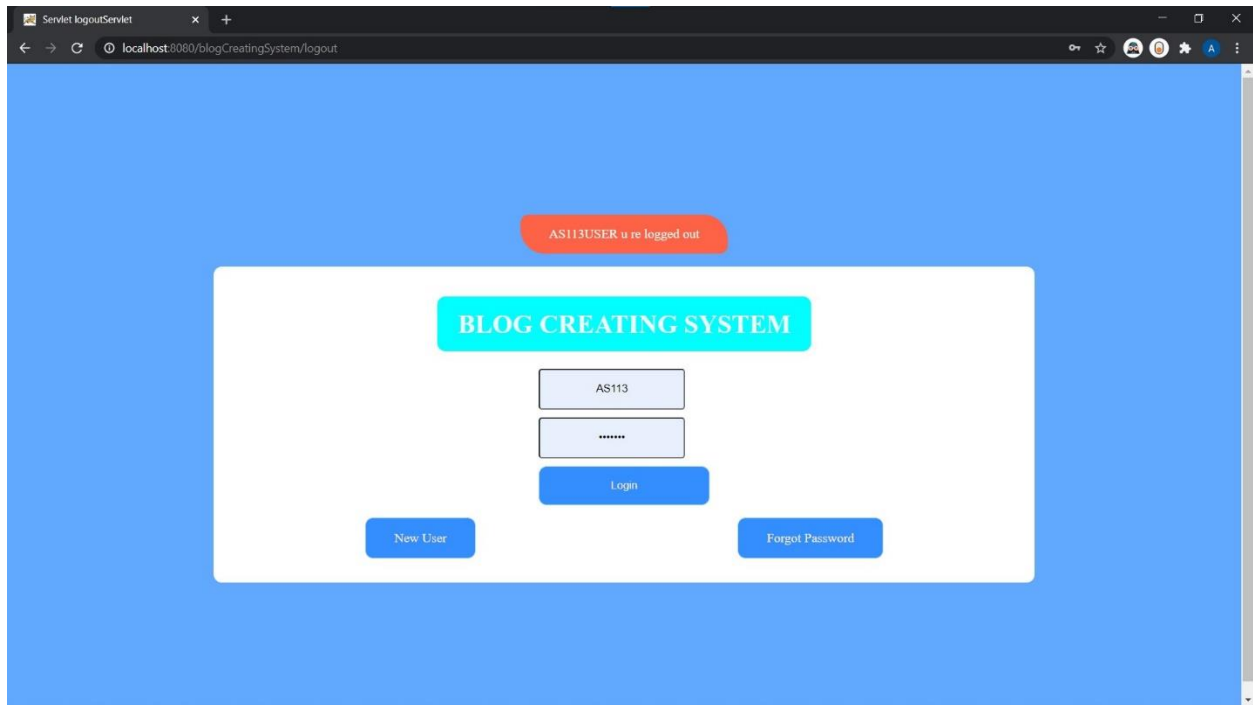




View profile



Logout



Documentation

Our blog creating system is dynamic in nature we can easily add features whenever we want

Through out the creating process one thing we comprehended clearly is we have to develop a model which is able to accept the changes i.e modules should independent so that if we want to make changes we don't want to collapse the whole structure of system

Our implementation went as developing the loosely coupled modules first and building the tightly coupled modules on the top loosely coupled modules .

Future work

We are planning to add features like content type search, content type categorization, recommending blog to user as personalized feed using ml algorithms

