

A
PROJECT REPORT ON

TwitBuzz Web Application

By

AYUSH PATEL (CE-001) (20CEUON118)
JENIS DONDA (CE-036) (20CEUOS070)

B.Tech CE Semester-VI
Subject: System Design Practice

Guided by:

Dr. Brijesh S. Bhatt
Professor
Dept. of Comp. Engg.



Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University



**Faculty of Technology
Department of Computer Engineering
Dharmsinh Desai University**

CERTIFICATE

This is to certify that the practical / term work carried out in the subject of
System Design Practice and recorded in this journal is the
bonafide work of

**AYUSH PATEL (CE-001) (20CEUON118)
JENIS DONDA (CE-036) (20CEUOS070)**

of B.Tech semester **VI** in the branch of **Computer Engineering**
during the academic year **2022-2023**.

Dr. Brijesh S. Bhatt
Professor,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Dr. C. K. Bhensdadia,
Head,
Dept. of Computer Engg.,
Faculty of Technology
Dharmsinh Desai University, Nadiad

Index

1.Abstract	1
2.Introduction.....	2
3.Software Requirements Specifications.....	3
4.Design	10
Usecase Diagram.....	10
Class Diagram.....	11
SequenceDiagram	12
Activity Diagram.....	13
Data Flow Diagram.....	15
Data Dictionary	18
5.Implementation Details.....	20
6.Testing	21
7.Screenshots.	22
8.Conclusion	29
9.Limitations and Future Extensions	30
10.Bibliography.....	31

Abstract

The aim of the project was to create a platform that allows users to share short messages with each other, follow other users, and engage in discussions on a variety of topics. The app was built using modern web technologies such as HTML, CSS, JavaScript, and React. The backend was implemented using Node.js and a MongoDB database.

The key features of the app include a user authentication system, the ability to create and edit user profiles, a Tweet feed that displays the latest tweets from other users, the ability to like and retweet posts, and a search function to find users and posts by keywords (like #trending). The app was designed to be responsive and mobile-friendly, with a simple and intuitive user interface.

Throughout the development process, various tools and techniques were employed to ensure the app's quality, including unit and integration testing, , and user feedback. The project was completed successfully, and the resulting app is a functional and user-friendly platform for sharing and engaging in conversations on a wide range of topics.

Introduction

Social media platforms have become an integral part of our daily lives, providing us with a means of sharing information, connecting with others, and engaging in discussions on a variety of topics. Twitter is one such platform that has gained widespread popularity due to its ability to deliver short, real-time messages to a global audience.

Inspired by the success of Twitter, we set out to develop a web application that would offer similar functionality while addressing some of the shortcomings of the platform. Our goal was to create a user-friendly and engaging platform that would allow users to share short messages, follow others, and engage in discussions on a variety of topics.

This project report details the development process of our TwitBuzz web application, highlighting its key features and functionalities. We believe that our application offers a compelling alternative to existing social media platforms, providing users with a fast, simple, and engaging platform for sharing and discussing information on a variety of topics.

2.1 Technology Used :

- Node JS
- Express JS
- React JS
- MongoDB
- HTML , CSS and Bootstrap

2.2 Tools Used :

- Visual Studio Code
- React Developer Tools
- Mongoose

Software Requirement Specifications

Functional Requirements:

The software needs to support three categories of functionalities as mentioned below:

1. Manage Users :

R.1.1 Register User

Description: To register in the system user needs to enter the username, password and email. This will be stored in the database of the system.

Input: User details.

Output: Email validation through OTP.

R.1.2 Login User

Description: If the user is already registered then the user can login in the system otherwise it will show the user not found message.

Input: Username and Password

Output: Login confirmation

R.1.3 Logout User

Description: If the user is already logged in then user can logout from the system.

Input: Click on logout button

Output: Logout successfully

R.1.4 Forgot Password

Description: Users can update the password with their email address. password reset link send via email to user on their registered email id.

Input: New Password

Output: redirect to login page

2. Manage tweets

R.2.1 Post a tweet

Description: A registered user would be able to post a tweet in the form of text, photos or links which will be visible to all users. Tweets can also have hashtags (eg. #indvspak).

Input: Tweet message.

Output: Confirmation message.

R.2.2 Like a tweet

Description: Whenever a user posts a tweet all his/her followers will be able to like that tweet and number of likes would be updated at that time. It would also be possible to unlike an already liked tweet.

Input: User Selection.

Output: Updated number of likes.

R.2.3 Comment on a tweet

Description: Whenever a user posts a photo all his/her other users will be able to comment on that tweet.

Input: Input Text.

Output: Comment posted.

R.2.4 Retweet a tweet

Description: Users would be able to retweet a tweet posted by any of users, this tweet can now be viewed by all users.

Input: Selection.

Output: Confirmation message.

R.2.5 View tweet with specific hashtag

Description: A user would be able to view all the tweets related to a specific hashtag. And also search a trending topics of tweets.

Input: User Selection for particular trend.

Output: List of tweets for that trend.

R.2.6 Delete a tweet

Description: A user who has posted a tweet would be able to any of his/her tweet.

Input: User selection.

Output: tweet is deleted.

R.2.7 Edit a tweet

Description: A user who has posted a tweet would be able to edit of his/her tweet.

Input: updated tweet message.

Output: tweet is edited.

3. Manage Profile

R.3.1 Post/Update Profile photo

Description: user can post/update his/her profile photo.

Input: Select photo from device .

Output: Profile photo updated.

R.3.2 Create/Update bio

Description: User can update/create his/her bio which consist birth Date of user, location and other message of user.

Input: new/updated bio credentials.

Output: Updated bio.

R.3.3 Follow/Unfollow user

Description: User can follow another user by click on follow button which is on profile of other user. And also unfollow user by clicking on following button.

Input: User Selection.

Output: button state change on user input.

R.3.4 View tweets and followers of user

Description: User can see other user's tweets and followers in their Profile. And also view profile of their followers.

Input: User Selection.

Output: list of tweets/list of followers.

4. Find users

R.4.1 Search Users

Description: User can search other users by their username in explore Section.

Input: User Search.

Output: list of users with matched text which is in search box.

R.4.2 View Profile

Description: based on search result user can visit particular user's Profile.

Input: User Selection.

Output: Profile of that particular user.

5. Manage Bookmarks

R.5.1 Add Bookmark

Description: User can bookmark a tweet for see again in future.

Input: Click on bookmark .

Output: tweet is added in bookmarks.

R.5.2 View Bookmarks

Description: User can see his/her bookmark tweets in bookmark section.

Input: Bookmarks section.

Output: list of bookmark tweets.

R.5.3 Remove from Bookmarks

Description: User can remove particular tweet from his/her bookmarks.

Input: Click on remove button.

Output: tweet is remove from bookmarks.

Non-Functional Requirements:

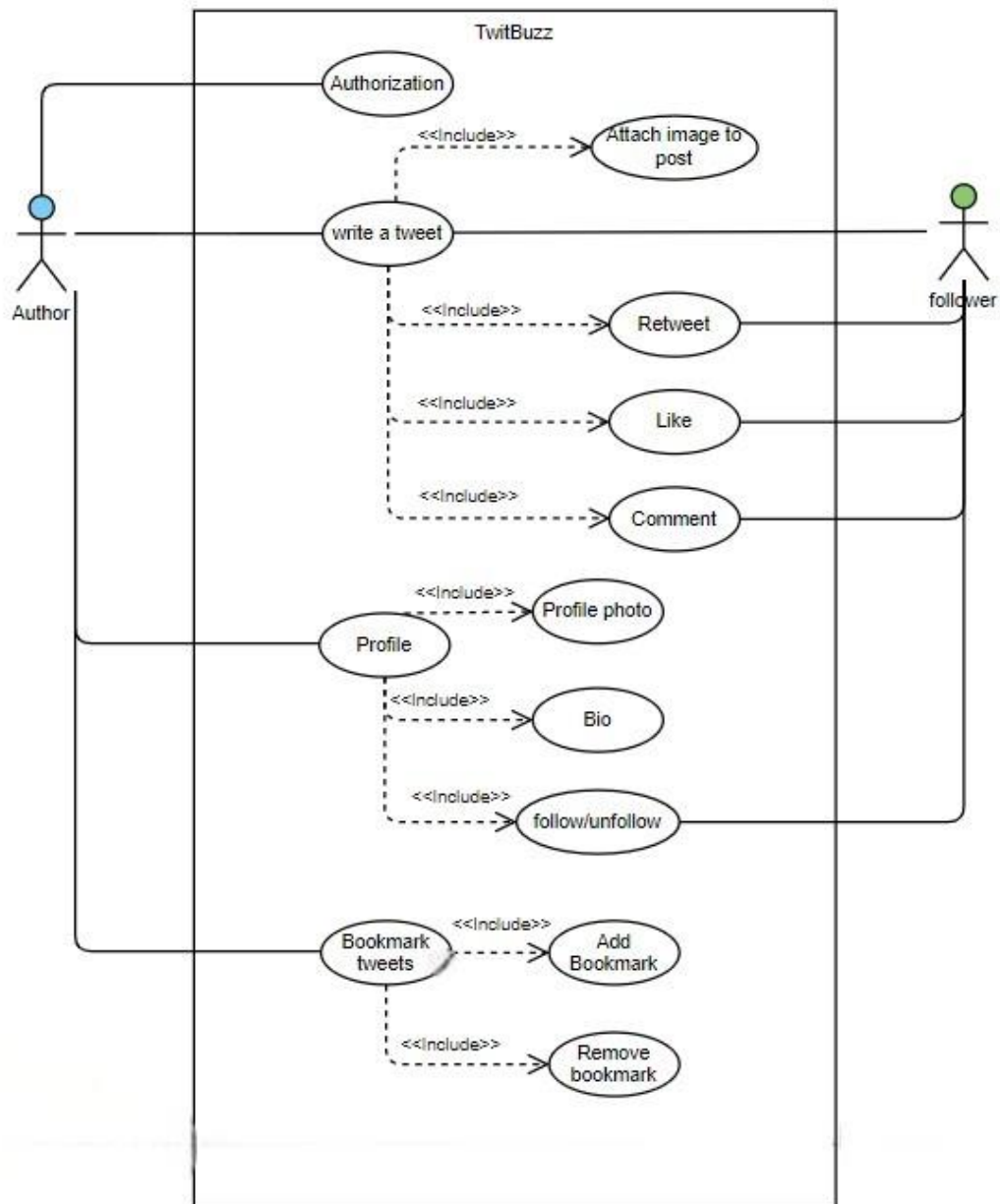
N.1: Database: A database management system that is available free of cost in the public domain should be used.

N.2: Generation of unique OTP: Once the user registered his/her email address verified through OTP number.

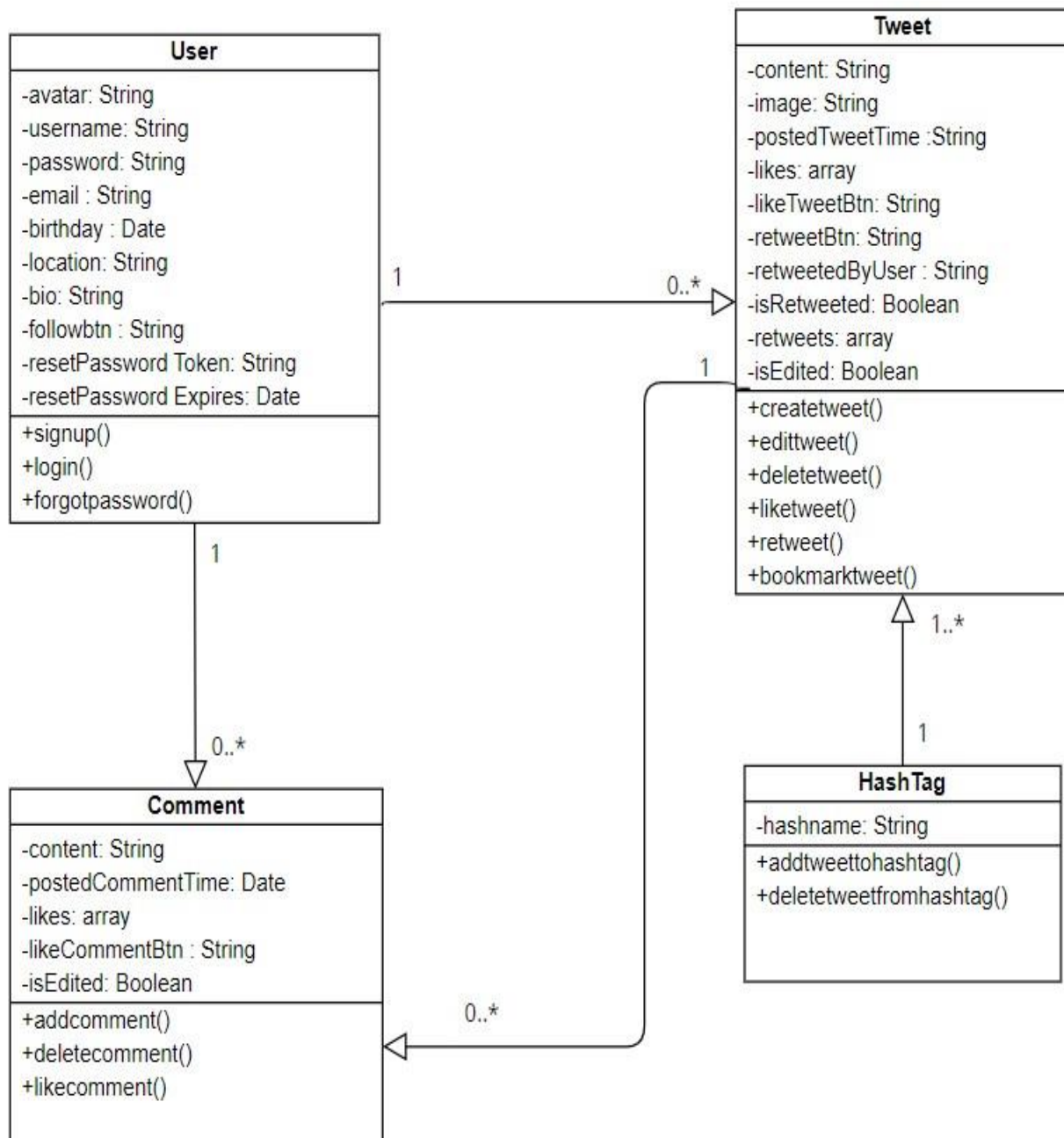
N.3: Responsivity : This web app is fully responsive. It can be easily used in mobile, tablets and ipad etc.

Design

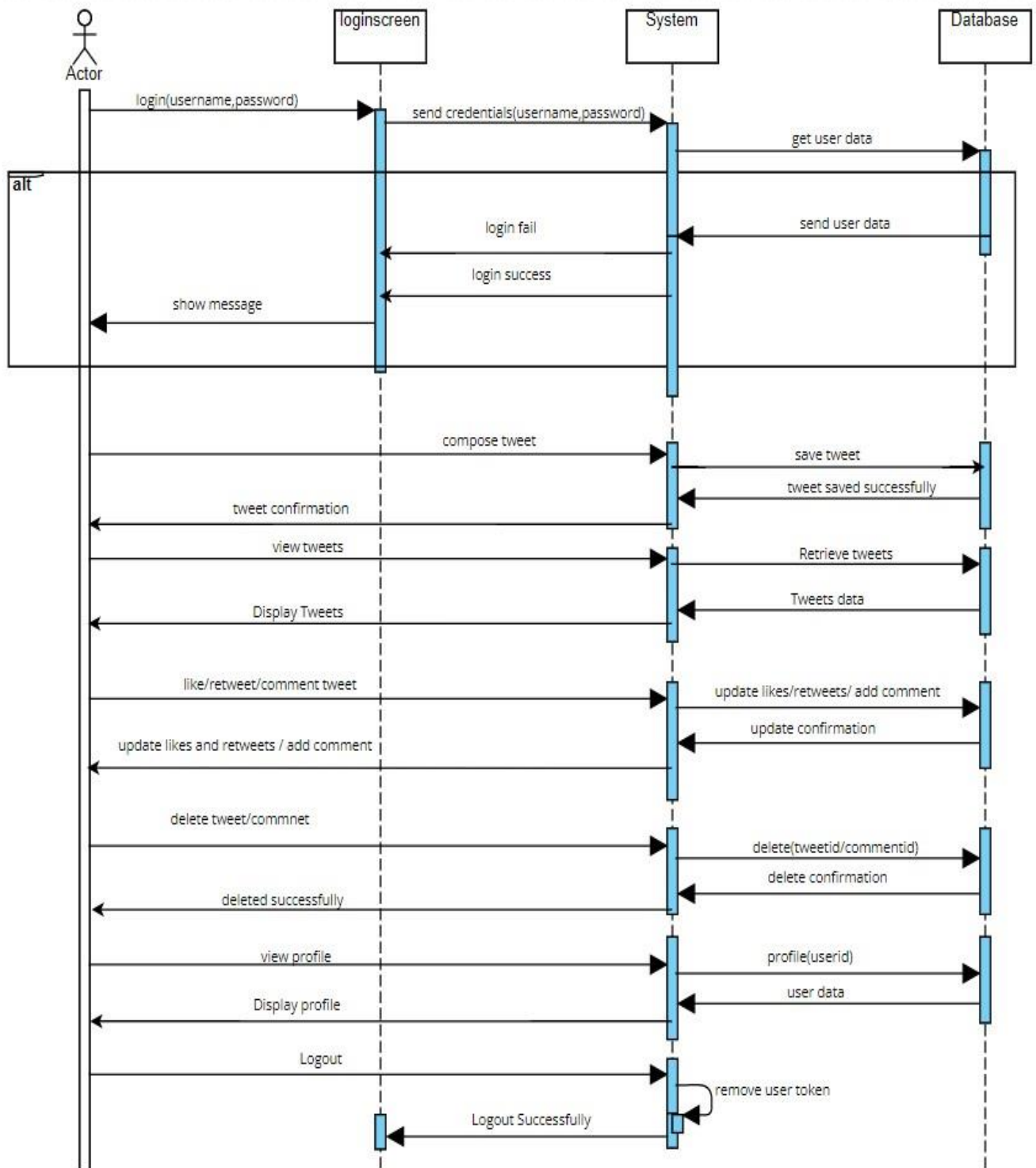
4.1 Usecase Diagram :



4.2 Class Diagram :



4.3 Sequence Diagram :



4.4 Activity Diagram :

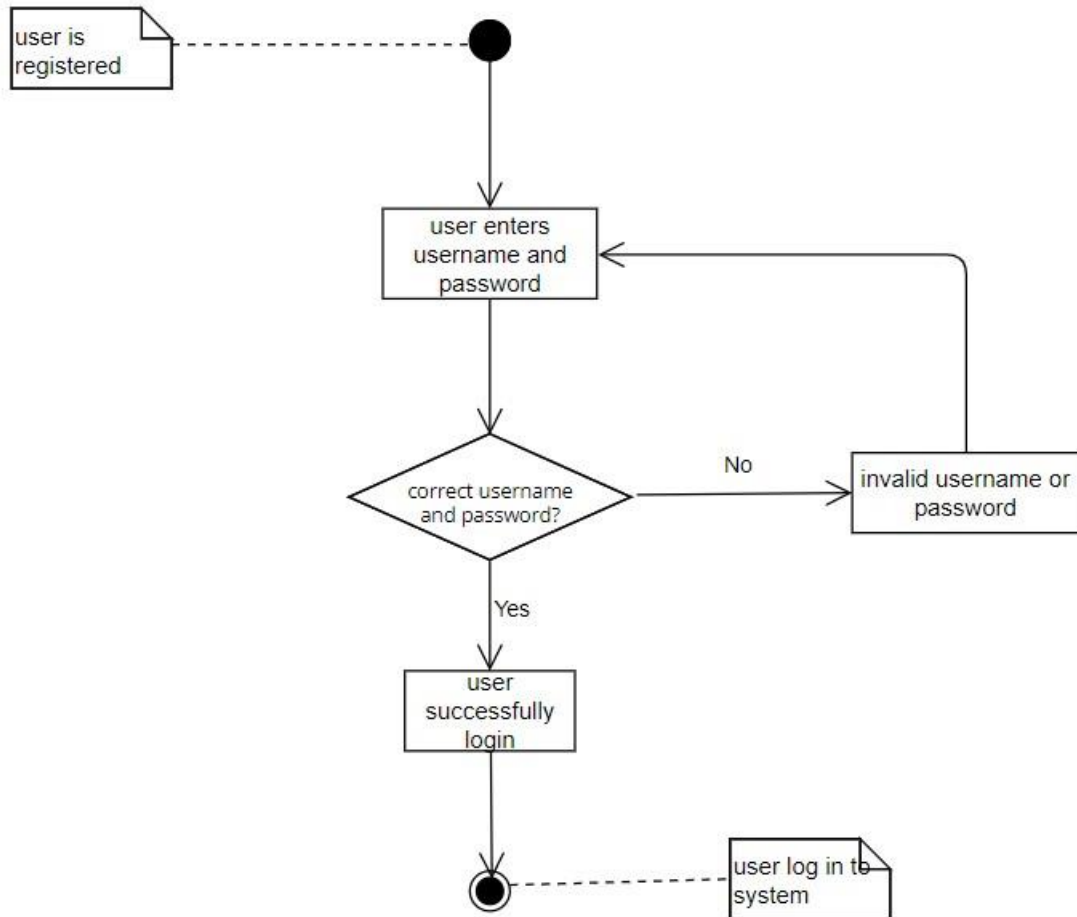


Fig 4.4.1 Activity diagram for user login

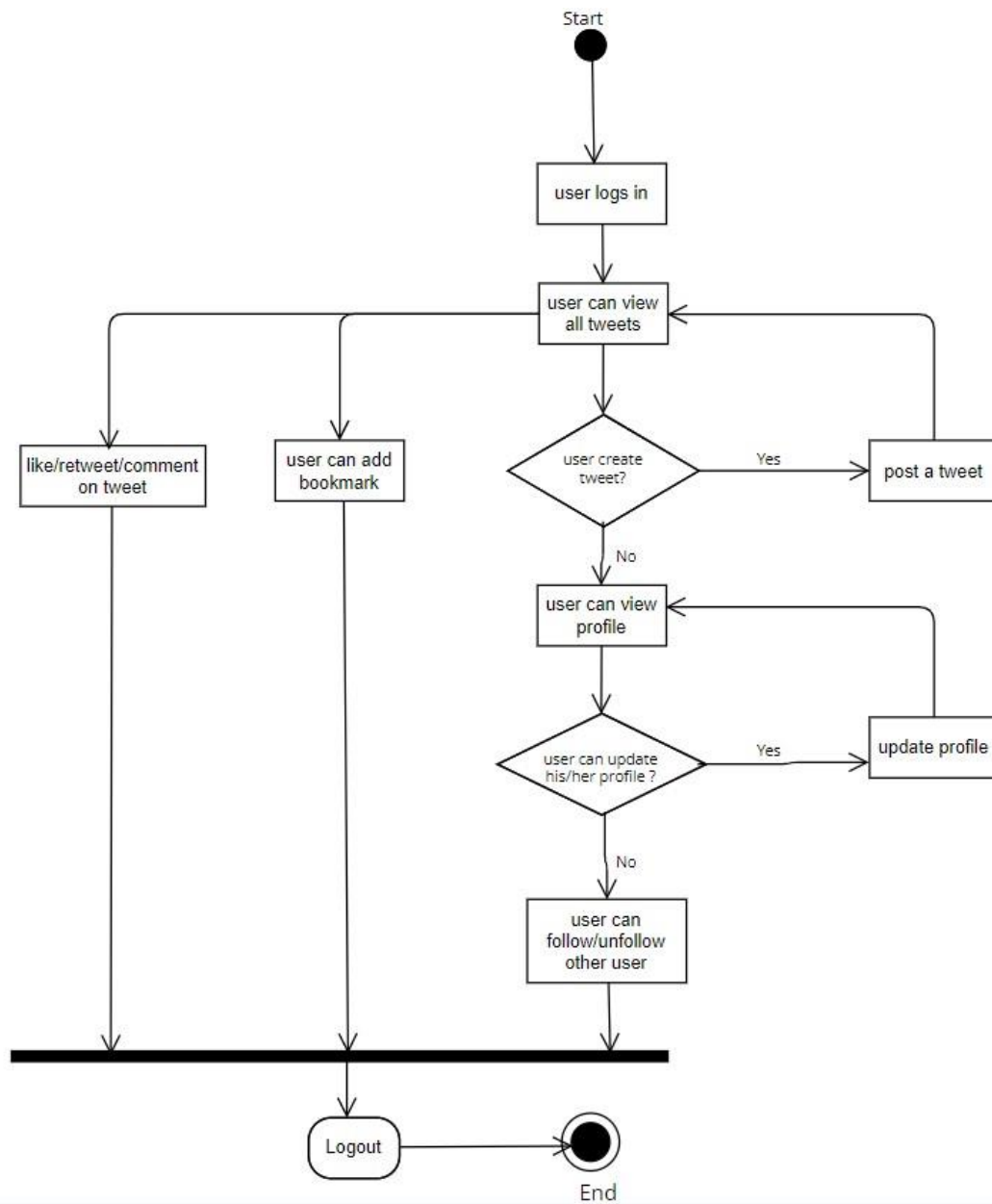
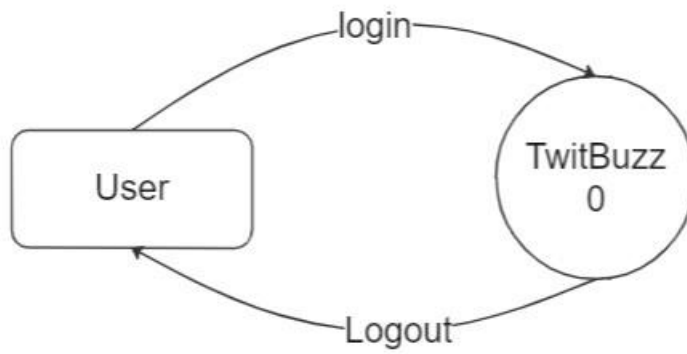
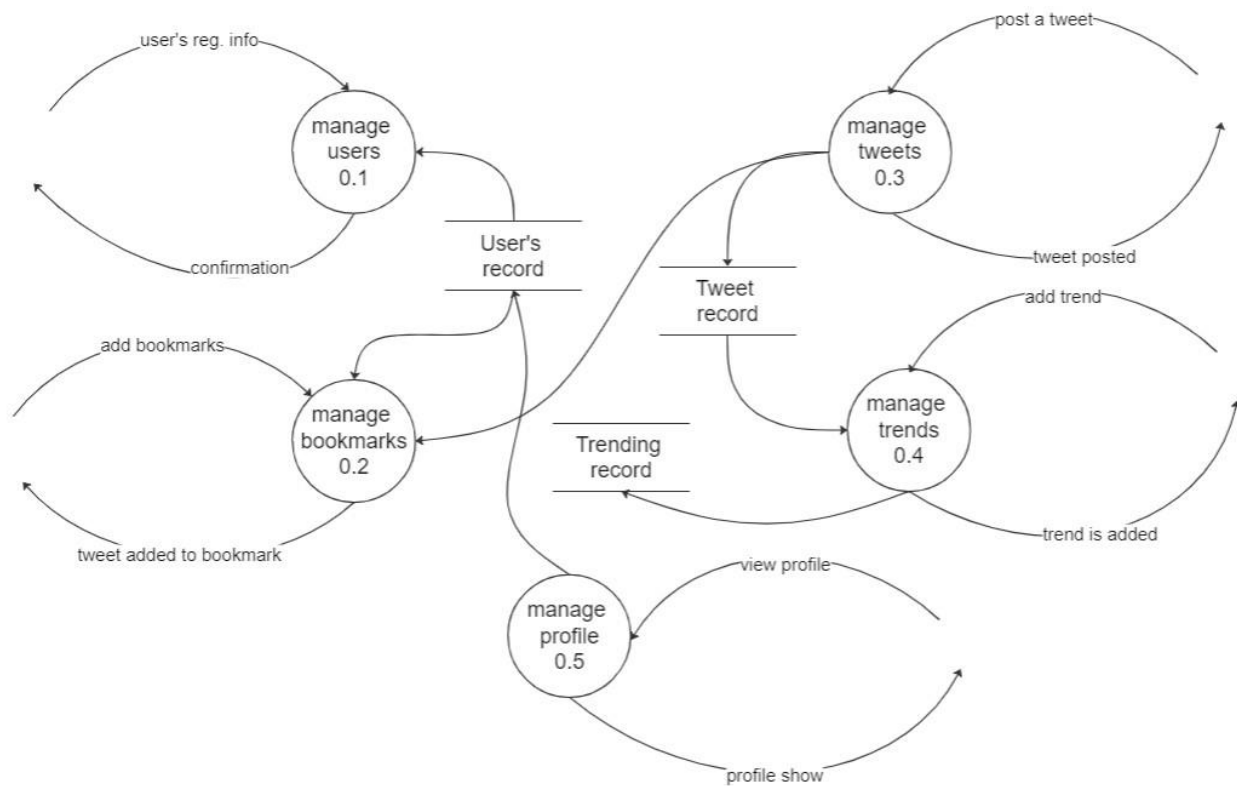


Fig 4.4.2 Activity diagram for TwitBuzz system

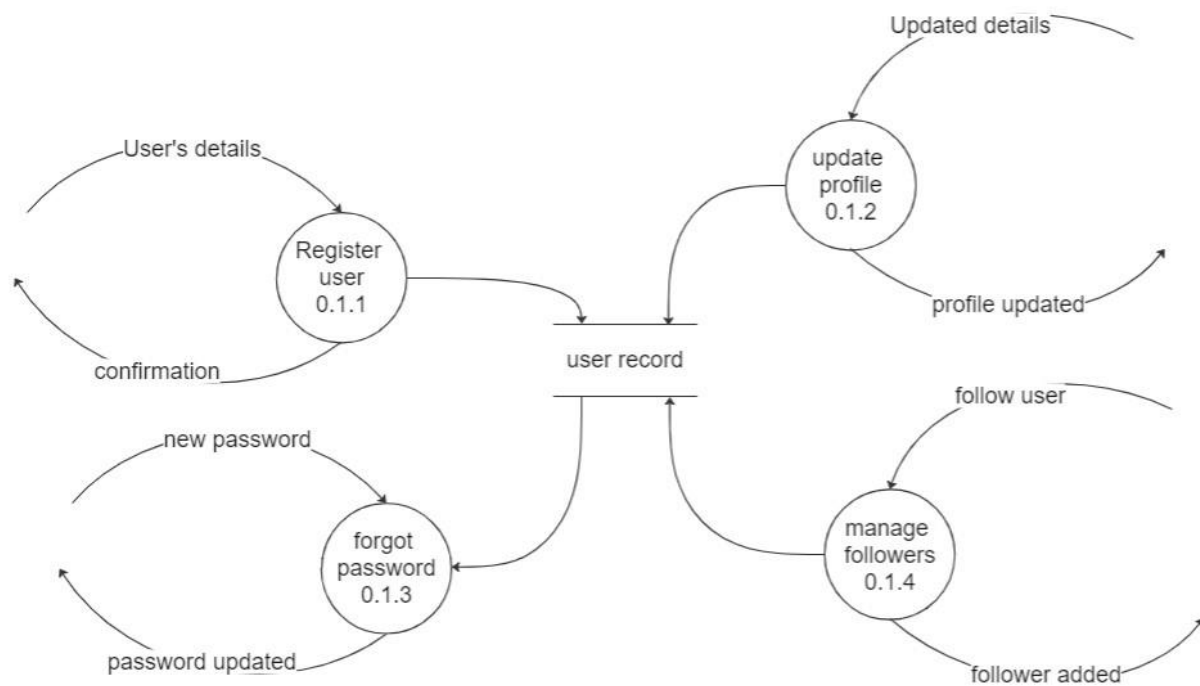
4.5 Data flow Diagram :



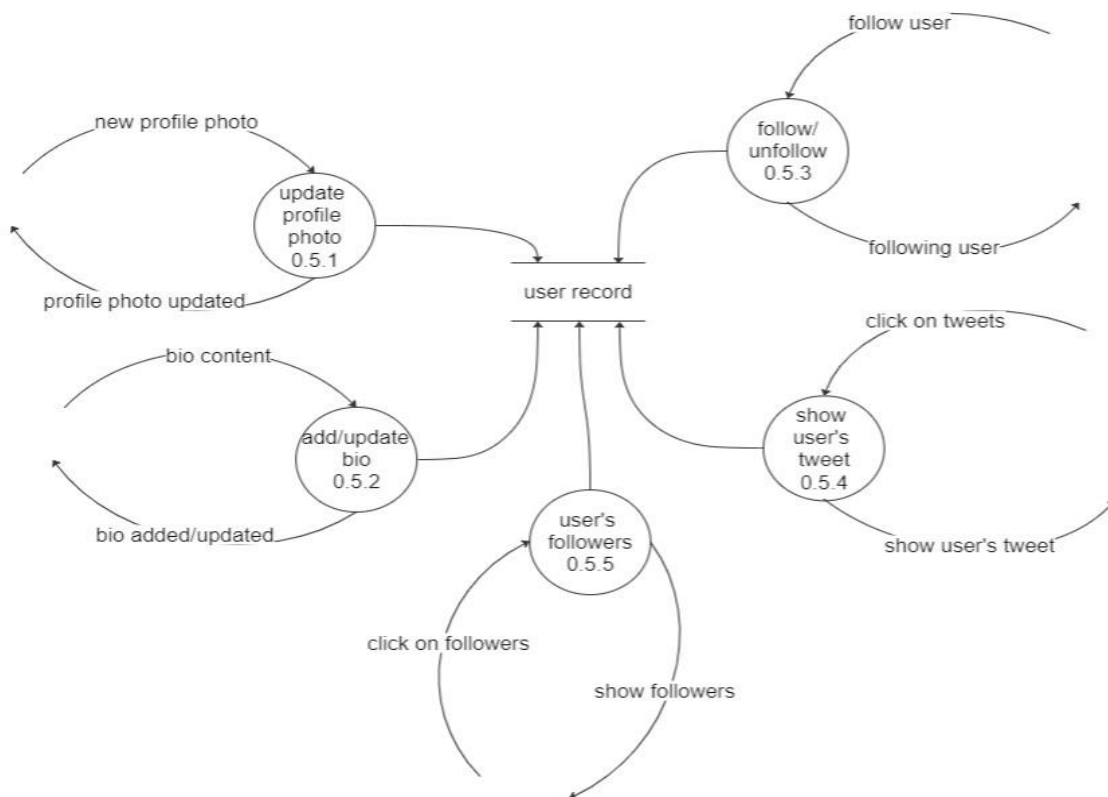
Level-0 DFD

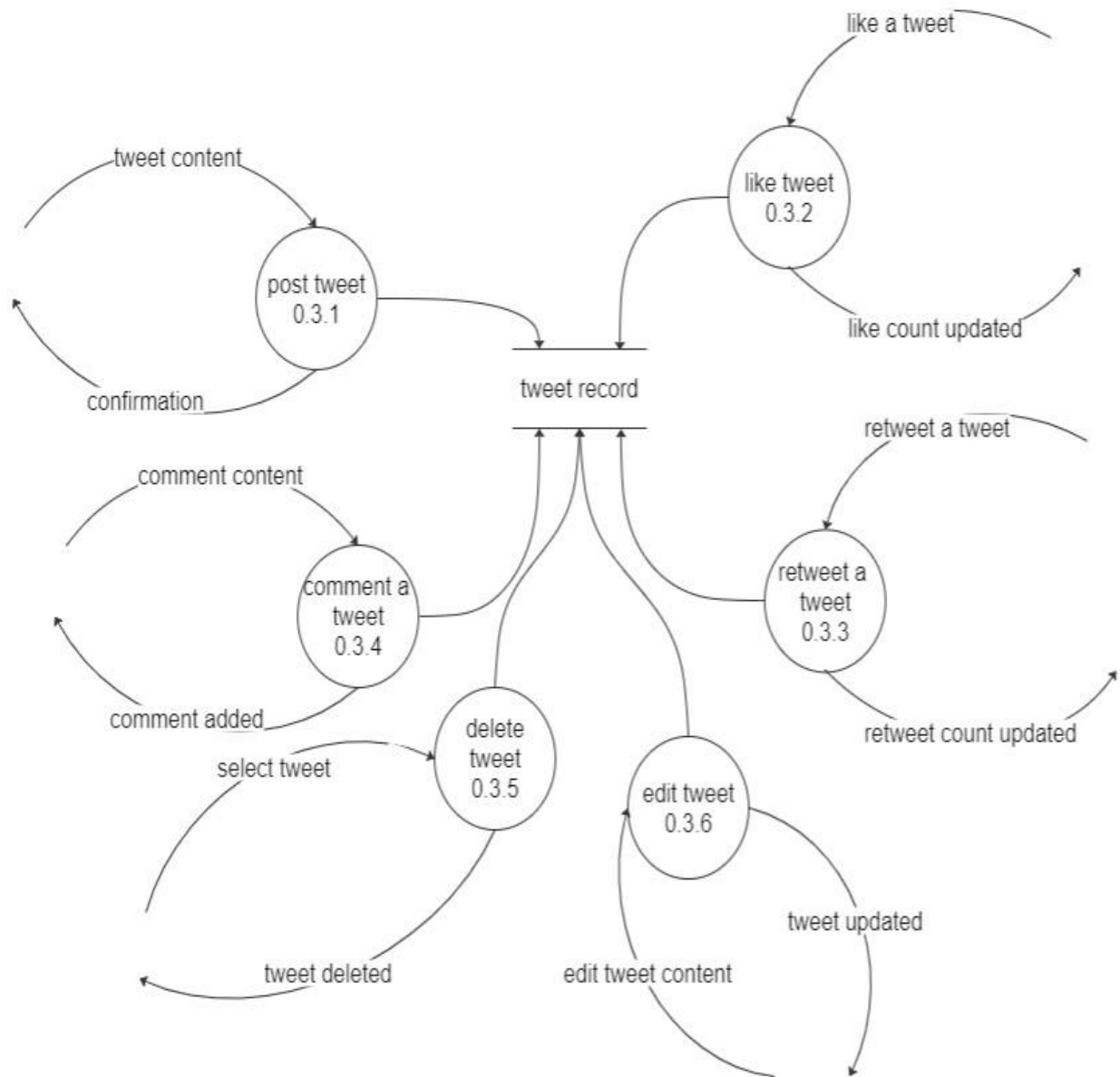


Level -1 DFD



Level -2 DFD





Level -2 DFD

4.6 Data Dictionary

❖ User Table :

Field name	Data type	Constraint	Description
avatar	String	-	Profile photo of user
username	String	Primary key	Username of user
password	String	-	Password of user account
email	String	-	Email id of user
followers	Object	-	Followers of user
followBtn	String	-	Show status of follow/following
birthday	Date	-	Birthday of user
location	String	-	Location of user
bio	String	-	Bio of user
resetPasswordToken	String	-	Token for reset password
resetPasswordExpires	Date	-	Time for expiring token
tweets	Object	-	Tweets of user
bookmarks	Object	-	Bookmarks of user

❖ HashTag table :

Field name	Data type	Constraint	Description
hashname	String	Primary key	Trend name
tweets	Object	-	Tweets of that particular hashname

❖ Tweet table :

Field name	Data type	Constraint	Description
content	String	-	Content of tweet
image	String	-	Image for tweet
postedBy	Object	-	User who post tweet
postedTweetTime	String	-	Time when tweet was posted
likes	array	-	Name of users who likes a tweet
likeTweetBtn	String	-	State change when user liked tweet
retweetBtn	String	-	State change when user retweeted tweet
retweetedByUser	String	-	User who retweet a tweet
isRetweeted	Boolean	-	True if tweet is retweeted else false
retweets	array	-	Array of retweets
isEdited	Boolean	-	True if tweet is edited else false
comments	Object	-	Comments on tweet

❖ Comment table :

Field name	Data type	Constraint	Description
content	String	-	Content of comment
postedBy	Object	-	User who post a comment
postedCommentTime	String	-	Time when Comment was posted
likes	array	-	Name of users who likes a comment
likeCommentBtn	String	-	State change when user liked a comment
isEdited	Boolean	-	True if tweet is edited else false

Implementation Details

● Modules

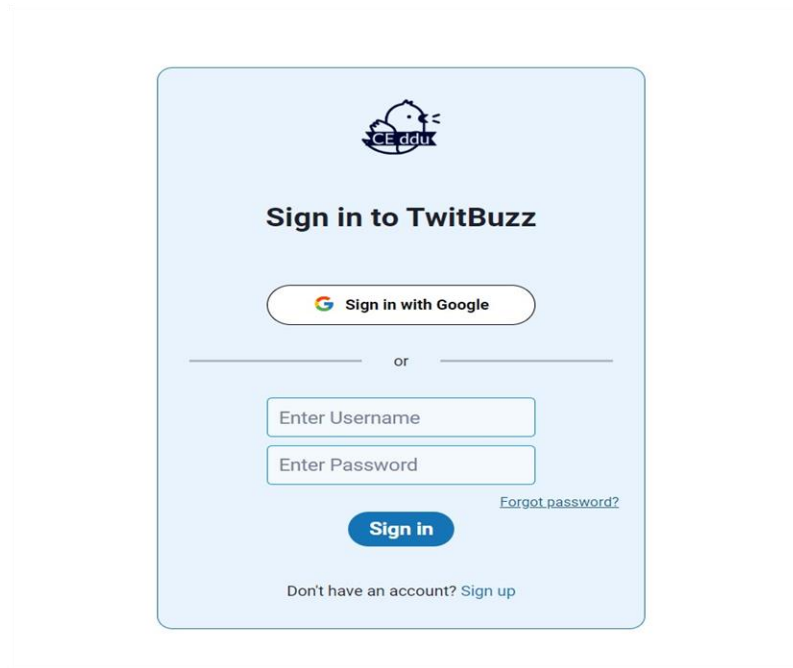
- Authentication : User have to register his/her self to the twitBuzz. And verify email id via email. And after that he/she can login to the system.
- Show all tweets : after login into system user can see all tweets that are posted by other users.
- Post a tweet : user can post a tweet which can contain text and image also.
- Like/retweet/commnet on tweet : user can like and retweet on particular tweet. And also commet on particular tweet. User can also like commnets of other users on particular tweet.
- Edit/delete a tweet : user can edit/delete his/her tweets.
- Trends : tweet which contains a '#' sign word those tweets are considered as hashed tweets. And '#' sign word as trend. So user can search trends and see those tweets which belongs to that particular trend.
- Update Profile : user can change his/her profile photo. User can add/update their bio which contains birthdate, location and message bio.
- Search user profile : user can search other users with their username and see their profiles. And also see that user's tweets and followers.
- Follow/unfollow : user can follow or unfollow (already followed) a particular user in their profiles.
- Bookmark a tweet : user can bookmark tweets. And also remove from bookmarks.
- Forgot password : user can reset password with reset password link which is shared to user via his/her registered email id.

Testing

- Testing Method : Manual Testing

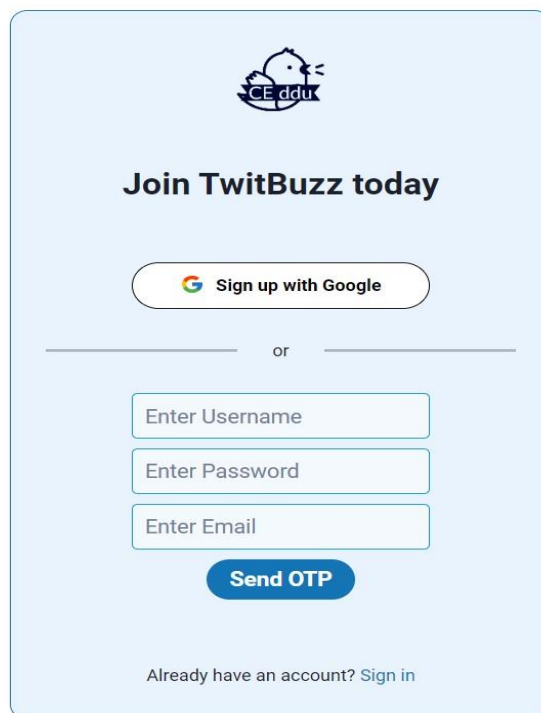
Sr No.	Test Scenario	Expected Results	Actual Result	Status
1.	Invalid Credentials for Login/Signup	Error	Error	Pass
2.	Invalid details at API call	Redirect back with error response	Redirect back with error response	Pass
3.	Invalid Redirect URI	Redirect to Error page	Redirect to Error page	Pass
4.	Valid API call (for All API calls)	Returns Response with success message	Returns Response with success message	Pass
5.	Invalid data in profile bio	Error	Error	Pass
6.	Invalid Email for forgot password	Error	Error	Pass
7.	Invalid trend name	Trend is not added	Trend is not added	Pass
8.	User session Expired	Logout automatically	Logout automatically	Pass

Screenshots



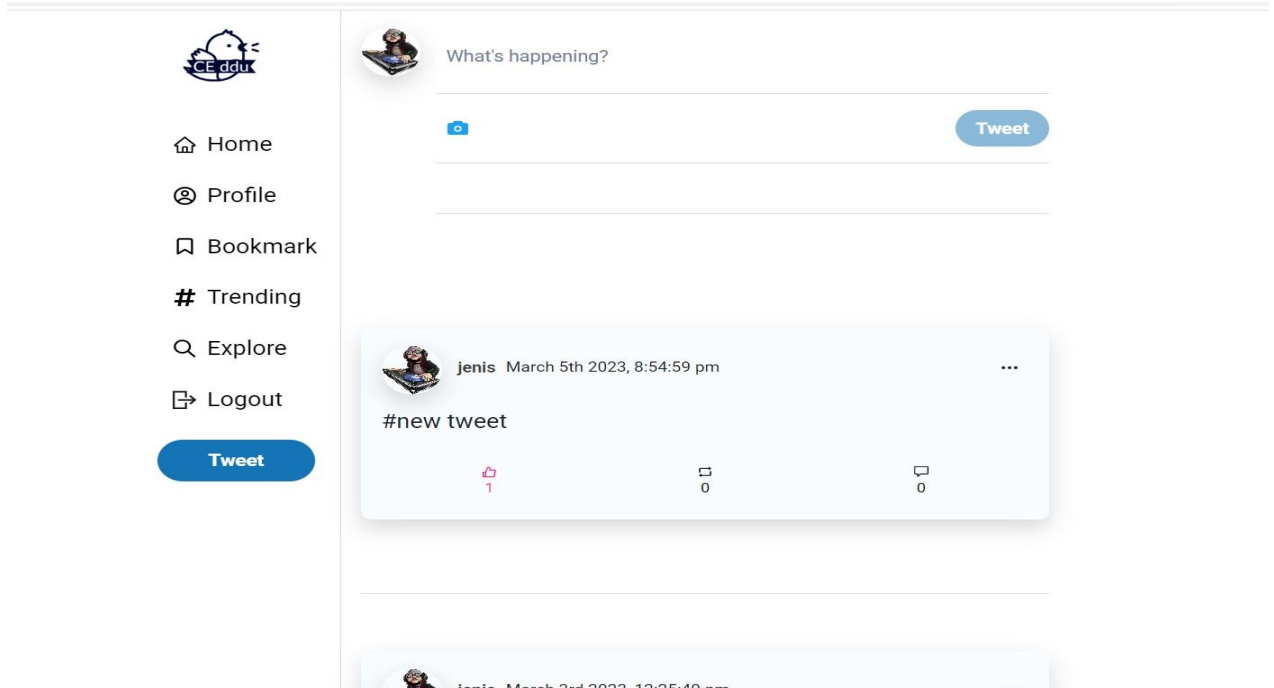
The screenshot shows a login interface for TwitBuzz. At the top is a logo featuring a bird and the text 'CE ddu'. Below the logo is the heading 'Sign in to TwitBuzz'. There is a 'Sign in with Google' button with the Google logo. A horizontal line with the word 'or' in the center separates this from the standard login fields. These fields include 'Enter Username' and 'Enter Password' text boxes. To the right of the password box is a link that says 'Forgot password?'. Below the password box is a blue 'Sign in' button. At the bottom of the form is a link that says 'Don't have an account? Sign up'.

7.1 Login page

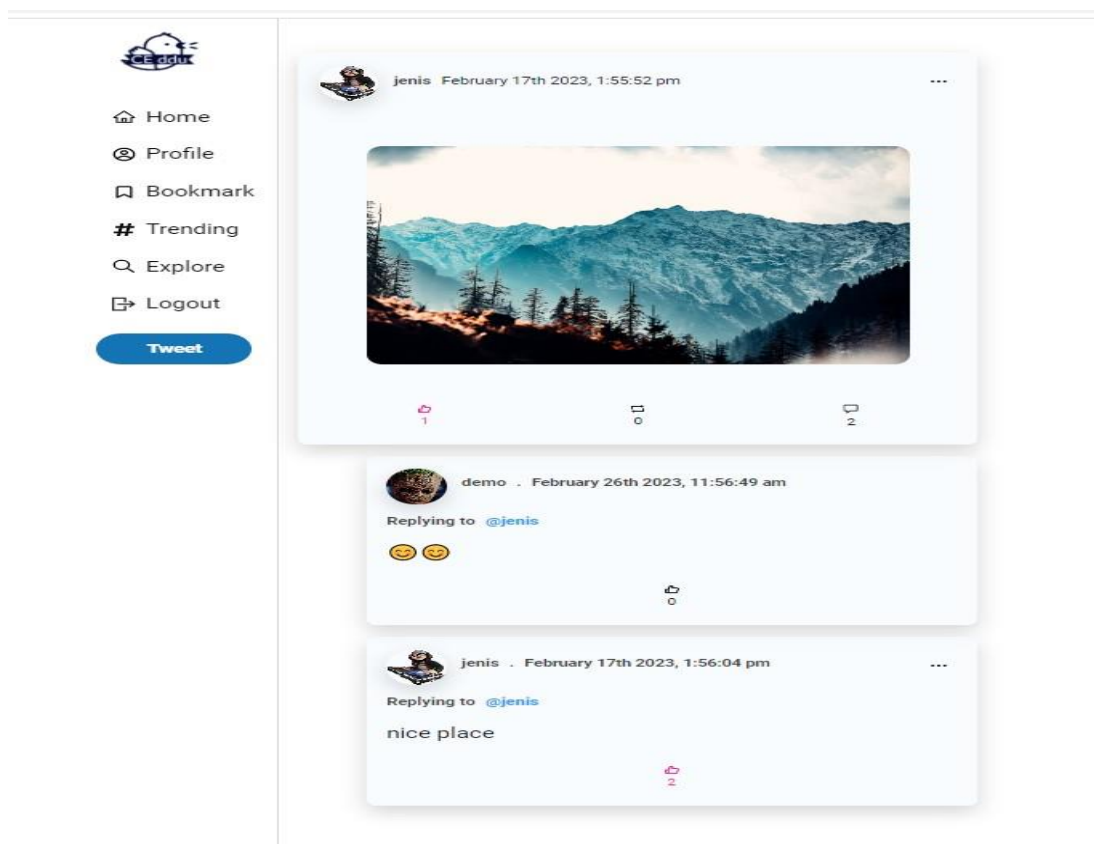


The screenshot shows a signup interface for TwitBuzz. At the top is the same 'CE ddu' logo. Below it is the heading 'Join TwitBuzz today'. There is a 'Sign up with Google' button with the Google logo. A horizontal line with the word 'or' in the center separates this from the standard signup fields. These fields include 'Enter Username', 'Enter Password', and 'Enter Email' text boxes. Below the email box is a blue 'Send OTP' button. At the bottom of the form is a link that says 'Already have an account? Sign in'.

7.2 Signup page



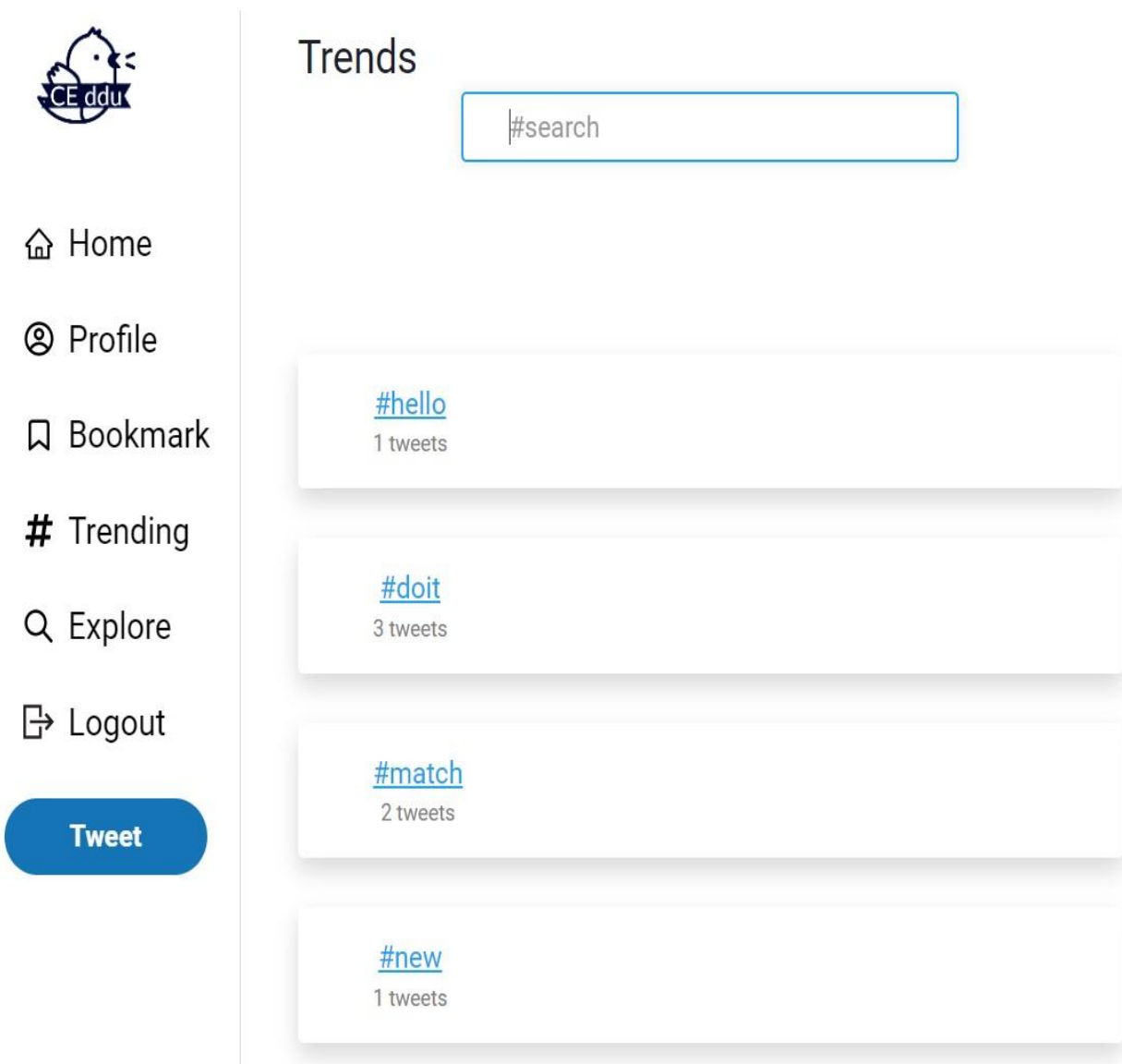
7.3 Home page



7.4 Comments on tweet



7.5 Edit/delete/bookmark tweet

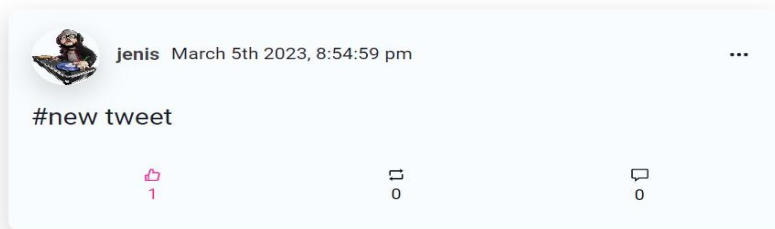


7.6 Trending page



- Home
- Profile
- Bookmark
- # Trending
- Explore

#new



7.7 #new trend's tweets



- Home
- Profile
- Bookmark
- # Trending
- Explore
- Logout

Tweet



Upload profile picture

Computer Engineering 
10/11/2002
Surat, Gujarat

Edit Bio

Enter Birthday
10/11/2002

Add Location
Surat, Gujarat

Enter Bio
Computer Engineering 

Save

Cancel

jenis

Tweets Followers

Tweets

jenis Retweeted



jenis February 25th 2023, 11:58:43 am

#match day  

7.8 current user profile



- Home
- Profile
- Bookmark
- # Trending
- Q Explore
- Logout

Tweet

Users



jenis

View Profile



demo

View Profile



ayush

View Profile

7.9 Explore user page



- Home
- Profile
- Bookmark
- # Trending
- Q Explore
- Logout

Tweet



Student 😊
10/05/2005
New york, USA

demo

Tweets Followers

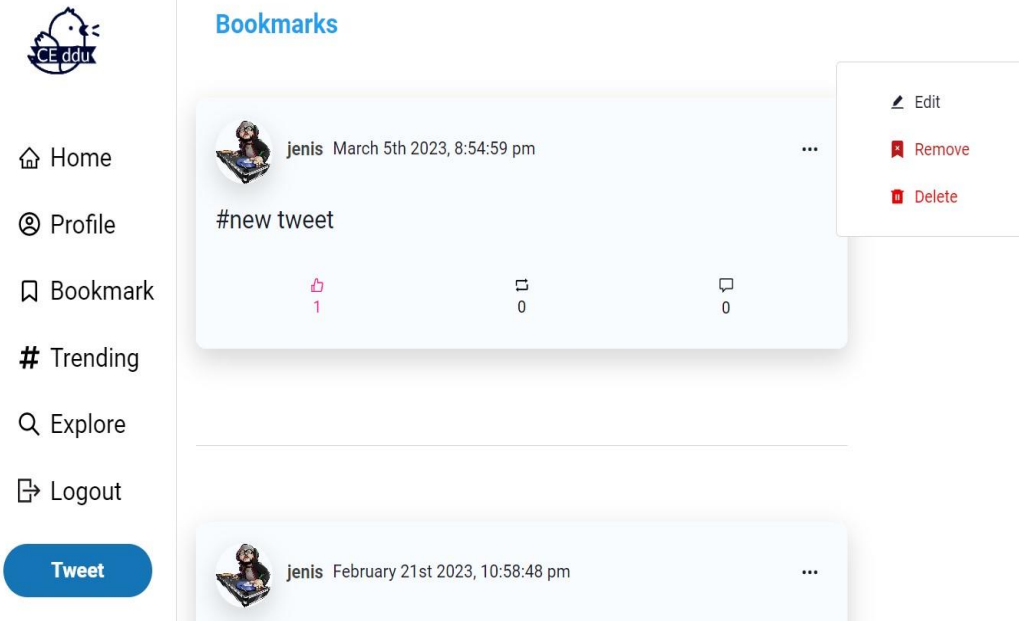
Following

followers

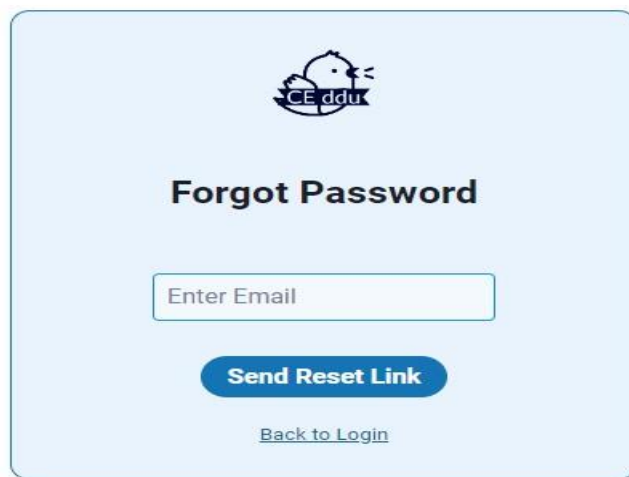


jenis

7.10 other user profile page



7.11 Bookmark tweet page



7.12 forgot password page

Hi jenis,

You are receiving this email because you (or someone else) has requested a password reset for your account.

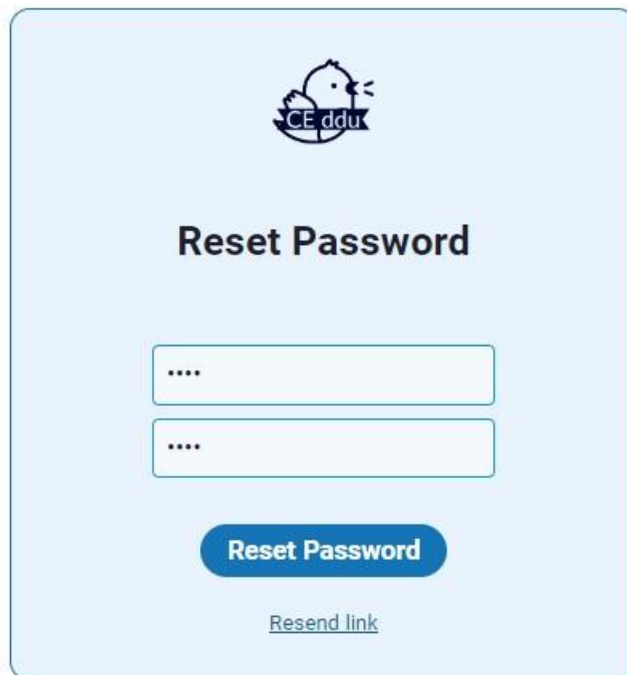
Please click on the following link, or paste this into your browser to complete the process:


<http://localhost:3000/reset-password/b689300aee2592ef56cb64853ef24275ce9fa669>

This link will be expired after 2 minutes.

If you did not request this, please ignore this email and your password will remain unchanged.

7.13 Reset password link on email

A light blue rounded rectangle containing a logo at the top center. The logo is a stylized bird with a banner that says 'CE ddu'. Below the logo, the text 'Reset Password' is centered in a bold, black font. Underneath this text are two input fields, each containing four dots to represent masked characters. Below the input fields is a blue button with the text 'Reset Password' in white. At the bottom of the form is a link labeled 'Resend link' in a smaller, blue font.



Reset Password

....

....

Reset Password

[Resend link](#)

7.14 Reset Password Page

Conclusion

The development of the TwitBuzz using MERN stack was a challenging but rewarding experience. The project aimed to create a platform that allows users to post, share and interact with tweets, and we were able to achieve this goal by implementing several features such as user authentication, tweet posting and liking.

This project provided a valuable learning experience for our team, as we gained practical knowledge in using the MERN stack and web development. We would like to express our gratitude to our project advisors and team members for their support and contribution to the project.

In conclusion, the development of the TwitBuzz using MERN stack was a rewarding experience, and we are proud of what we have accomplished. We hope that this project will serve as a valuable resource for those interested in learning more about web development and the MERN stack.

Limitations and Future Extensions

9.1 Limitations

- Tweets are can not be classified topic wise.
- Follow other users suggestions are not there.
- Dark mode is not be implemented in TwitBuzz.

9.2 Future Extensions

- We will be trying to overcome our limitations.
- Also use of some advanced algorithm to classified tweets topic wise.

Bibliography

- <https://legacy.reactjs.org/>
- <https://stackoverflow.com/>
- <https://nodejs.org/en>
- <https://www.mongodb.com/docs/>