

# Project Wazzup/COSC 360

**Wasek Habib – 51201144**

**Introduction:** Wazzup is a social networking website with a view to creating positive and friendly community, where the users/students can see other users' recent stories and support each other to fight mental illness. The fully functional website is located at -

<http://project360-com.stackstaging.com/>

**Executive Summary:** In brief, apart from viewing and searching for stories and profiles without registering, the users can sign up by using an email and password. After logging in, they should be able to

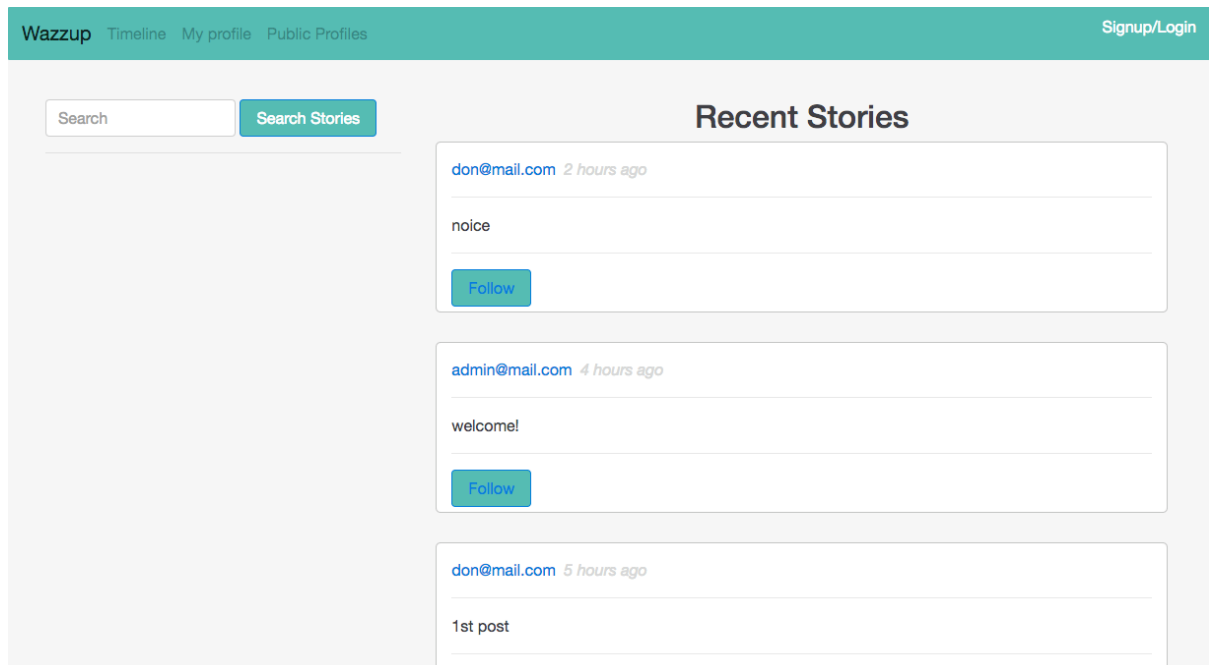
- post stories semi-anonymously (i.e. with their email)
- follow/unfollow each other
- view both public and their friend's (after following) stories
- see public profiles
- search for stories/users

Some of the other key features of the website:

- Responsive and mobile friendly
- Server-side scripting with PHP
- Data storage in MySQL
- Appropriate security for data (i.e. SQL injection prevention, salt hashing etc.)
- Site maintains state (user state being logged on, etc.)
- AJAX utilization for asynchronous updates for login/signup, following, and posting stories
- User profiles stored in database
- Activity by date (latest stories are displayed on the top)
- Error handling
- Created with MVC architecture

Note: Milestone 1 was created with hand-styled layout with custom made responsive and mobile friendly framework and form validation with jQuery, but due to time constraints and some technical difficulties, Bootstrap has been used for the later part.

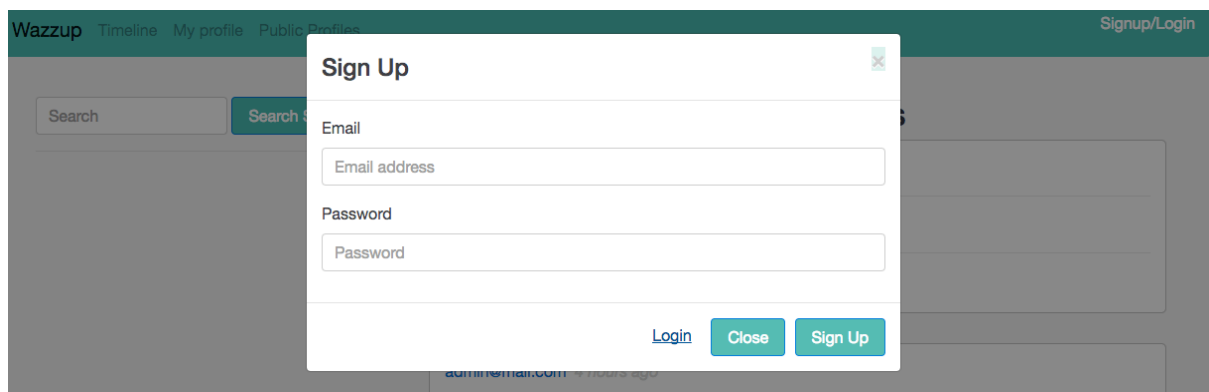
### Walkthrough:



*Figure: Homepage*

**Guest:** A guest can see the public stories by clicking on the logo or timeline. However, he won't be able to perform other activities (i.e. posting stories, follow) except checking out the public profiles and stories associated with them.

**Sign up:** Registration can be done by clicking on the Signup/Login button on the right of the navigation bar. He will see a modal/popup login form with email and password field.



After clicking on the “Sign up” link, he will see the above registration form. After signing up he will be directed to the homepage and see a logout option.

**Logout:** After clicking logout he will see the signup/login option again.

**Login:** Login process is similar to the sign up process.

**Viewing public stories:** Users can view recent public stories on the homepage.

**Post stories:** after posting something in post story (max 200 letters) form, he will see an alert box saying whether the story has been posted successfully or not. Please note in order to see the story he has to refresh the page since AJAX had not been used for retrieving/displaying stories.

**Follow/unfollow:** It can be performed by clicking the follow button. After following the user will have the option to unfollow.

**View all friends’ story:** The user has to go to Timeline for this feature.

**View a user’s profile/story:** It can be performed by clicking on the user’s id (email).

**View own profile/story:** The user has to go to My profile for this feature.

**Search public profiles:** The user has to go to Public profiles to see the active profiles.

**Search stories:** It can be performed by searching keywords.

**File Descriptions:** The website was created with MVC type architectural style. The view folder contains all necessary files for the interface and database is the model.

**views/header.php:** Contains the code for navigation bar and the components inside it.

**views/home.php:** Contains code for the homepage and public stories.

**views/footer.php:** Contains code for footer and modal.

**views/timeline.php:** Contains the code for displaying stories of the people who the user is following.

**views/yourtweets.php:** Contains the code for displaying user’s own profile or stories.

**views/publicprofile.php:** Contains the code for showing public/active profiles.

**views/search.php:** contains the code for displaying search results of story keywords.

**index.php:** Includes the pages user wants.

**functions.php:** Contains the main logic of the app. Connects with the database and keep state (i.e. session). Includes displayTweets, displaySearch, displayTweetBox, displayUsers,

and `time_since` functions. The `displayTweet` function is written in a way that the user has to follow someone first to see his friend's activities in his timeline. Otherwise, the timeline will show public stories.

***actions.php***: Works as the API. Can be called to post stories and login function. Contains server side validation.

***styles.css***: Contains necessary stylesheet.

***scripts.js***: Contains necessary javascript files.

***wazzup-3541db.sql***: Contains the SQL code for database model.

#### **Known Limitations:**

- No client side validation for this version.
- Users can follow/unfollow their own profile.
- No AJAX for displaying stories. The user has to refresh the page after posting to see his story.
- No admin panel due to time constraint.