Test what I have so far

* make sure like/spam buttons work everywhere with a value in them
* test

Votes:

'n' : not concluded

's' : yes, she likes me!

'f' : no, sigh...

'x' : skipped story

Databases:

1. User
   1. **Username, (maybe max 50 chars) (primary key)**
   2. Password (maybe max 50 chars)
   3. Name (maybe max 50 chars)
   4. Age (must be an int)
   5. Gender/Sex (M/F/Other)
   6. Score
   7. Right
   8. Wrong guesses
   9. Location
   10. Email (maybe max 100 chars)
   11. Date of registration
   12. Current story which user is on
   13. Current location
2. Stories Users Have Seen Before
   1. **Username (both primary key)**
   2. **Story ID**
   3. Date
3. Stories and Tags
   1. **Story ID (both primary key)**
   2. **Tag (50 chars)**
4. Comments and Likers
   1. **Comment ID (both primary key)**
   2. **Username**
   3. Date
5. Stories and Likers
   1. **Story ID (both primary key)**
   2. **Username**
   3. Date
6. Comments and Spam
   1. **Comment ID (both primary key)**
   2. **Username**
   3. Date
7. Story and Spam
   1. **Story ID (both primary key)**
   2. **Username**
   3. Date
8. Stories (maybe split into new, done, and bad)
   1. **Story ID (primary key)**
   2. Title (50 char max?)
   3. Actual story (1000 char max?)
   4. User who wrote it (foreign key)
   5. Date/Time written
   6. Date/Time concluded, if applicable
   7. Concluded yet (Y/N/Incomplete)
   8. Like count
   9. Spam count
   10. User comment (1000 char max, for the conclusion)
   11. Number of yes votes
   12. Number of no votes
   13. Location (state)
9. Comments (maybe split into good/bad)
   1. **Comment ID (primary key)**
   2. Text (say 1000 char max?)
   3. Date/Time written
   4. Spam count
   5. Like count
   6. User who wrote it (foreign key)
   7. Story to which it belongs (foreign key)
10. Stories and Pictures (maybe in file system??)
    1. Story ID
    2. Picture
11. UUID
    1. uuid
    2. date created
    3. date expires
    4. expired
    5. used
    6. username
    7. password
    8. email
12. UUIDResetPwd
    1. uuid
    2. email
    3. new password
    4. date created
    5. date expires
    6. expired
    7. used
13. Users and votes
    1. Story ID
    2. Username
    3. Yes/No
    4. Date

Features:

1. Login
   1. Keep logged in
   2. Email confirmation to prevent spam accounts
   3. Forgot username/pwd
   4. Demographics
      1. *Username* (maybe max 50 char) (unique)
      2. *Password* (maybe max 50 char)
      3. Name (maybe max 50 char)
      4. Age (must be an int)
      5. Gender/Sex (M/F/Other)
      6. Score initialized to 1000
      7. Right guesses initialized to 0
      8. Wrong guesses initialized to 0
      9. Location (State)
      10. Date of registration
      11. *Email* (maybe max 50 char) (unique)
2. Create Story
   1. Create title (maybe 50 char max?)
   2. Actual story (maybe 1000 char max)
   3. Tags (each tag must be less than 50 chars, less than 50 tags)
   4. Location (optional, a US state)
   5. Picture (multer will need to be changed to multipart)
3. Review Own Stories
   1. Update your own story text
   2. Sort stories by date created (oldest first, to incentivize them filling out what happened)
   3. View comments on their story by others
      1. Report and like comments
      2. Delete comments
   4. Conclude your story
      1. Yes or no
      2. Possible to add a comment to conclude
4. Review other stories
   1. Yes, No, Skip
   2. Filter stories
   3. Report and like stories
   4. Report and like comments
   5. After voting
      1. See Y/N split of others
      2. View comments on this story
         1. Report and like comments
      3. Move onto next story
5. User data
   1. Only reset password
   2. See rank

Misc

1. Make sure user can’t see their own story on their feed
2. Allow change usernames
3. Change color scheme
4. Maybe change user locations later
5. Login with FB and Google later??
6. Points added for guessing correctly, incorrectly, posting a story, and concluding a story (based on number of reviews too)
   1. Lost points for spam comments and stories
   2. Eventually account will get deleted
7. If server crashes, restart automatically
   1. Make sure the ID’s are actually unique!!
8. Emoji support and iOS support and Android support
   1. If not above, make sure at least the stuff works on mobile
9. See stories in order
   1. This can be changed later on
   2. For example, to sort by likes, I can have little bots in the database and switch every time they see an inversion. They keep cycling.
   3. Precompute!!
10. How does passport change?
11. Get https verified
    1. Buy SSL key
12. Delete your account
13. Delete stories/comments which have too many spam votes
14. Day limit to update stories
    1. Else it gets deleted and maybe a points penalty
15. No negative score!
16. Save partially filled in responses
17. Where do you need res.end()
18. See older stories?
19. Also maybe call it TwooHoo
20. Maybe a streak feature like snapchat
21. Is bcrypt deprecated?
22. Check max size of server you can send through POST and stuff like that
23. XSS and Directory Traversal and Stack Overflow and DDoS attack monitoring
24. Friends feature?
25. Prevent double clicking
26. change all to html 5 templates
27. http://stackoverflow.com/questions/218907/how-to-handle-multiple-submissions-server-side
28. http://www.the-art-of-web.com/javascript/doublesubmit/
29. ACCEPT LARGER SIZES IN HTML/EJS FORMS, MYSQL DATABASE, AND IN THE CODE FOR VERIFICATION IN ROUTE AND STUFF LIKE THAT
30. MAKE SURE EVERYTHING WORKS ASYNCHRONOUSLY
31. EMAIL NOT AT GMAIL.COM
32. GOTTA SWITCH EMAIL PROVIDERS SOMETIME IN THE FUTURE
33. Change the links to actual ones--see comments in the code, and do testing
34. Change required="true" to required="required"
35. Maybe the HTML get/post requests are not correct after changing (from /changePassword to /settings/changePassword...etc) in the redirect stuff
36. ALL COMMENTING STUFF ISN'T IMPLEMENTED YET;
37. can't like own story????
38. block friends from seeing (via Facebook login)
39. add delete story in update/conclude
40. set default tags on update conclude, also update tags
41. maybe mysql varchar 50 isn't enough to store a length 50 string
42. Update points when the story got concluded, and make it so you can't "unconclude" a story
43. rewrite the get story code for review stories to use jquery to inject the necessary data
44. refresh likes/spam after someone likes/spam
45. also make it so you can only click the button once
46. click again to "unlike/unspam"
47. create index of functions and comments
48. delete all res.ends which are redundant

**right now, the right/wrong in the user database doesn't work--** for update/conclude, update all users which guessed correctly and incorrectly, and then all the scores--this is what I need to do

1. clean up code
2. COMMENTS
3. make sure all the functions in the post fail or all succeed to prevent people from upping their like counts by sending http requests they made
4. What is the difference between the above and below divs!? in reviewStories.ejs
5. escape MySQL statements to make sure that even if username is ;drop data then it's alright
6. make sure resultsJS.js has everything it needs (didn't get accidentally deleted)
7. Battle friend
   1. New stories and possibly the best of old stories