

Totem

Sprint 1: Design Inspection, Code Inspection and Unit Testing

Team 20

Cameron Glass, Todd Griffin, Dominic Miller,
Nirali Rai, and Rishabh Ramsisaria

Design Inspection Defects (20)

Product	Navigation Bar		
Author	Cameron Glass, Todd Griffin		
Defect #	Description	Severity	How Corrected
1	Logout must be a feature that can be accessible from the Nav Bar on every page. We only had it on the settings page.	2	The logout button was added to the nav bar.
2	The links should be in a more natural order. Previous order: Personal Private Public Tournament	3	The links were arranged in a more natural order: Public Private Tournament Profile
3	The NavBar is only visible at the top of the page.	2	The NavBar was changed so that it is always visible at the top even after scrolling down.

Product	Sign Up page		
Author	Dominic Miller, Nirali Rai		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/signup/index.html	3	The links were set to the corresponding directory. The url required now would be:com/signup/

3	No restriction was set to password length or otherwise.	2	Restriction set for password to be at least 8 letters
4	When one of the fields is invalid the type of invalidity is not specified	2	The invalidity would be specified, for example: - password needs to be at least 8 characters - email is not correct
5	No restriction was set on the username	2	Restriction set for username to be between 2 and 20 characters
6	No restriction was set on the display name	2	Restriction set for username to be between 1 and 30 characters

Product	Login page		
Author	Rishabh Ramsisaria		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/login/index.html	3	The links were set to the corresponding directory. The url required now would be:com/login/

Product	Forgot password Page		
Author	Cameron Glass, Todd Griffin		
Defect #	Description	Severity	How Corrected

1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/forgotpassword/index.html	3	The links were set to the corresponding directory. The url required now would be:com/forgotpassword/

Product	Splash page		
Author	Dominic Miller, Nirali Rai		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/splash/index.html	3	The links were set to the corresponding directory. The url required now would be:com/splash/

Product	Settings page		
Author	Rishabh Ramsisaria		
Defect #	Description	Severity	How Corrected
1	The name of this page was not	2	The name of the html page

	set to index.html so that the browser can directly open the required page when the link is clicked.		was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/settings/index.html	3	The links were set to the corresponding directory. The url required now would be:com/settings/

Product	Profile page		
Author	Dominic Miller, Nirali Rai		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	Each button was going to lead to a subpage but it was better to use modals to avoid excess coding as well as complexity.	2	Each button opens a modal while remaining on the profile page.
3	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/profile/index.html	3	The links were set to the corresponding directory. The url required now would be:com/profile/
4	This page was called personal page which is similar to private which signifies the private feed	3	The name was changed to profile page since it is easier to differentiate between different pages now.
5	Each user should be able to	1	Added a modal on the emoji on

	change their avatar(emoji) but the front end did not contain any means to.		the profile page allowing a user to select their avatar(emoji).
--	--	--	---

Product	Public page		
Author	Rishabh Ramsisaria		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The image 2 field for a challenge mustn't exist.	2	The image 2 field will be hidden in the create challenge modal.
3	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/public/index.html	3	The links were set to the corresponding directory. The url required now would be:com/public/
4	When a poll is created, we were going to ask for an end date as expiration for determining how long the poll was going to be active. This would however, hinder the request mechanism for a challenge poll since the end time would need to depend on when the poll was started and not what was specified since the start time would depend on when the request was accepted.	2	The create poll now requires duration in terms of days, hours and minutes.
5	When a request is made for polls to be displayed in the feed, 50 requests are returned.	2	The request returns 25 polls instead of 50 since this reduces load on client side.

6	The filter bar according to theme was a simple search bar but since we are hardcoding possible themes, we want the user to be able to see the options that can be used for filtering.	2	The search now has a drop down with the list of possible themes which changes as the user types in the search bar. Multiple themes can also be selected.
---	---	---	--

Product	Private Page		
Author	Rishabh Ramsisaria		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The image 2 field for a challenge mustn't exist.	2	The image 2 field will be hidden in the create challenge modal.
3	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/private/index.html	3	The links were set to the corresponding directory. The url required now would be:com/private/
4	When a poll is created, we were going to ask for an end date as expiration for determining how long the poll was going to be active. This would however, hinder the request mechanism for a challenge poll since the end time would need to depend on when the poll was started and not what was specified since the start time would depend on when the request was accepted.	2	The create poll now requires duration in terms of days, hours and minutes.
5	When a request is made for polls to be displayed in the feed, 50	2	The request returns 25 polls instead of 50 since

	requests are returned.		this reduces load on client side.
6	The filter bar according to theme was a simple search bar but since we are hardcoding possible themes, we want the user to be able to see the options that can be used for filtering.	2	The search now has a drop down with the list of possible themes which changes as the user types in the search bar. Multiple themes can also be selected.

Product	Tournament Page		
Author	Cameron Glass, Todd Griffin		
Defect #	Description	Severity	How Corrected
1	The name of this page was not set to index.html so that the browser can directly open the required page when the link is clicked.	2	The name of the html page was set to index.html
2	The page link was set to the html file itself which overcomplicates the url. For example to get to the login page, the url would be similar to:com/tournament/index.htm 	3	The links were set to the corresponding directory. The url required now would be:com/tournament/

Code Inspection Defects

Functions not defined or found in unit_testing files. Had to reformat method declaration. Severity: 3	
login(username, password){ if(verifyInput(username,	login = (username, password) => { if(verifyInput(username,

<pre> password)){ // move to next page return 1; } else{ return 0; } } </pre>	<pre> password)){ // move to next page return 1; } else{ return 0; } } </pre>
<pre> signup(username, displayName, email, password, passwordVerify){ if(verifyInput(username, displayName, email, passwordVerify, password)){ // move to next page // Createuser(username, displayName, email, password, passwordVerify); return 1; } return 0; } </pre>	<pre> signup = (username, displayName, email, password, passwordVerify) => { if(verifyInput(username, displayName, email, passwordVerify, password)){ // move to next page // Createuser(username, displayName, email, password, passwordVerify); return 1; } return 0; } </pre>
<pre> changeDisplayName (changedisplayname, repeatdisplayname){ if(verifychangeDisplayName(changedisplay name, repeatdisplayname)){ return 1; } return 0; } </pre>	<pre> changeDisplayName = (changedisplayname, repeatdisplayname) => { if(verifychangeDisplayName(changedisplay name, repeatdisplayname)){ return 1; } return 0; } </pre>
<pre> changePassword(changepassword, repeatpassword){ if(verifychangePassword(changepassword , repeatpassword)){ return 1; } return 0; } </pre>	<pre> changePassword = (changepassword, repeatpassword) => { if(verifychangePassword(changepassword , repeatpassword)){ return 1; } return 0; } </pre>
<pre> forgotPassword(username, email) { if(verifyInput(username, email)) { // move to login page return 1; } } </pre>	<pre> forgotPassword = (username, email) => { if(verifyInput(username, email)) { // move to login page return 1; } } </pre>

<pre> return 0; } </pre>	<pre> return 0; } </pre>
--------------------------	--------------------------

Modal overwrote the styling put on a button class to change the color. Had to assign importance to the class to make the color be what it is set to.
Severity: 3

<pre> .personal-opt{ color: #551a8b; font-size: 300%; /* text-align: center */ position: relative; } </pre>	<pre> .personal-opt{ color: #551a8b !important; font-size: 300%; /* text-align: center */ position: relative; } </pre>
<pre> .personal-opt:hover{ color: violet; font-size: 360%; } </pre>	<pre> .personal-opt:hover{ color: violet !important; font-size: 360%; } </pre>

Using # as comment in CSS instead of /**/ in profile page in classes and id's
Severity: 3

<pre> #emoji{ align-content: center; text-align: center; #border: 1px solid white; #border-radius: 25%; #height: 150%; width: 100px; margin-top: 1rem; margin-bottom: 1rem; } </pre>	<pre> #emoji{ align-content: center; text-align: center; /* #border: 1px solid white; */ /* #border-radius: 25%; */ /* #height: 150%; */ width: 100px; margin-top: 1rem; margin-bottom: 1rem; } </pre>
<pre> #emoji:hover{ align-content: center; text-align: center; #border: 1px solid white; #border-radius: 25%; #height: 155%; width: 115px; margin-top: 1rem; margin-bottom: 1rem; } </pre>	<pre> #emoji:hover{ align-content: center; text-align: center; /* #border: 1px solid white; */ /* #border-radius: 25%; */ /* #height: 155%; */ width: 115px; margin-top: 1rem; margin-bottom: 1rem; } </pre>

}	}
#displayName{ text-align: center; align-content: center; font-size: 1.5rem; #font-style: italic; font-weight: bold; margin: 0; color: white; }	#displayName{ text-align: center; align-content: center; font-size: 1.5rem; /* #font-style: italic; */ font-weight: bold; margin: 0; color: white; }
#username{ text-align: center; align-content: center; font-size: 1.5rem; #font-style: italic; margin: 0; color: white; }	#username{ text-align: center; align-content: center; font-size: 1.5rem; /* #font-style: italic; */ margin: 0; color: white; }
.stats{ text-align: center; align-content: center; font-size: 1.2rem; color: white; #border: 1px solid white; margin-top: 10px; border-radius: 20%; background-color: rgba(0, 0, 0, 0.10); padding: 5%; }	.stats{ text-align: center; align-content: center; font-size: 1.2rem; color: white; /* #border: 1px solid white; */ margin-top: 10px; border-radius: 20%; background-color: rgba(0, 0, 0, 0.10); padding: 5%; }

Margin bottom for buttons on profile page was preventing the page from resizing appropriately Severity: 2	
.btn-margin{ margin-bottom: 8vw !important; }	.btn-margin{ margin-bottom: 8vw !important; } #no-bottom{ /*applied to bottom buttons*/ margin-bottom: -10vw; }

There was no jwt secret set in .env would crash program

Security: 1

```
require('dotenv').config();
const app = require('express')();
const bodyParser = require('body-parser');
const fs = require('fs');
```

```
require('dotenv').config();
const app = require('express')();
const bodyParser = require('body-parser');
const fs = require('fs');

// make sure JWT_SECRET environment
variable is set
if (!process.env.JWT_SECRET ||
process.env.JWT_SECRET ===
'YOUR_JWT_SECRET') {
  console.log('Error: must copy
.env.sample => .env and then fill it out
(JWT_SECRET: string)');
  process.exit(1);
}
```

Program would not work correctly if production database was not created

Security: 2

```
// make sure JWT_SECRET environment
variable is set
if (!process.env.JWT_SECRET ||
process.env.JWT_SECRET ===
'YOUR_JWT_SECRET') {
  console.log('Error: must copy
.env.sample => .env and then fill it out
(JWT_SECRET: string)');
  process.exit(1);
}
```

```
// make sure JWT_SECRET environment
variable is set
if (!process.env.JWT_SECRET ||
process.env.JWT_SECRET ===
'YOUR_JWT_SECRET') {
  console.log('Error: must copy
.env.sample => .env and then fill it out
(JWT_SECRET: string)');
  process.exit(1);
}
```

```
// make sure the correct databases exist
and have been built
if (process.env.NODE_ENV ===
'production') {
  if (!fs.existsSync('./db/prod.db')) {
    console.log('Error: must create
production database to run production
server');
    console.log('run `npm run
```

	<pre>init-db-prod`); process.exit(1); } }</pre>
--	---

Program would not work correctly if development database was not created Severity: 2	
<pre>// make sure the correct databases exist and have been built if (process.env.NODE_ENV === 'production') { if (!fs.existsSync('./db/prod.db')) { console.log('Error: must create production database to run production server'); console.log('run `npm run init-db-prod`'); process.exit(1); } }</pre>	<pre>// make sure the correct databases exist and have been built if (process.env.NODE_ENV === 'production') { if (!fs.existsSync('./db/prod.db')) { console.log('Error: must create production database to run production server'); console.log('run `npm run init-db-prod`'); process.exit(1); } } else { if (!fs.existsSync('./db/dev.db')) { console.log('Error: must create development database to run development server'); console.log('run `npm run init-db-dev`'); process.exit(1); } }</pre>

On SQL query, not every item that was needed was there to create a user, so it needed to be checked Severity: 2	
<pre>const data = { email: req.body.email, username: req.body.username, display_name: req.body.display_name, hash: req.body.password, emoji: req.body.emoji, friend_challenges: 0, friend_challenges_won: 0,</pre>	<pre>// returns true if the given object contains all the given properties // object: {} , properties: [] function validateObject(object) { return Object.keys(object).every(p => { if (object[p] === 0) { return true; } }) }</pre>

<pre> tiki_score: 0, polls_created: 0 }; if (!utils.validateObject(data)) { res.status(400).send({ code: 400, info: 'malformed body parameters: email, username, display_name, password, emoji' }); return; } </pre>	<pre> return object[p]; }); } const data = { email: req.body.email, username: req.body.username, display_name: req.body.display_name, hash: req.body.password, emoji: req.body.emoji, friend_challenges: 0, friend_challenges_won: 0, tiki_score: 0, polls_created: 0 }; if (!utils.validateObject(data)) { res.status(400).send({ code: 400, info: 'malformed body parameters: email, username, display_name, password, emoji' }); return; } </pre>
---	--

<p>The JWT bearer token authorization was not set so the program would crash</p> <p>Severity: 1</p>	
<pre> if (!req.headers.authorization req.headers.authorization.split(' ').length !== 2 req.headers.authorization.split(' ')[0] !== 'Bearer') { return { code: 401, info: 'invalid auth header; Authorization: Bearer <JWT>' }; } req.jwt = jwt.verify(req.headers.authorization.split(' </pre>	<pre> try { if (!req.headers.authorization req.headers.authorization.split(' ').length !== 2 req.headers.authorization.split(' ')[0] !== 'Bearer') { return { code: 401, info: 'invalid auth header; Authorization: Bearer <JWT>' }; } req.jwt = </pre>

<pre>')[1], process.env.JWT_SECRET); return { code: 200, info: 'success' }</pre>	<pre>jwt.verify(req.headers.authorization.split(' ')[1], process.env.JWT_SECRET); return { code: 200, info: 'success' } } catch(e) { return { code: 401, info: 'invalid JWT token' }; }</pre>
--	---

<p>The JWT bearer token authorization was malformed so the program would crash</p> <p>Severity: 1</p>	
<pre>req.jwt = jwt.verify(req.headers.authorization.split(' ')[1], process.env.JWT_SECRET); return { code: 200, info: 'success' }</pre>	<pre>if (!req.headers.authorization req.headers.authorization.split(' ').length !== 2 req.headers.authorization.split(' ')[0] !== 'Bearer') { return { code: 401, info: 'invalid auth header; Authorization: Bearer <JWT>' }; } req.jwt = jwt.verify(req.headers.authorization.split(' ')[1], process.env.JWT_SECRET); return { code: 200, info: 'success' }</pre>

<p>Queries were sent for user profiles that do not exist which crashed the program</p> <p>Severity: 1</p>	
<pre>const result = await db('users') .insert({ email, username, display_name,</pre>	<pre>const result = await db('users') .insert({ email, username, display_name,</pre>

<pre> emoji, hash, friend_challenges, friend_challenges_won, tiki_score, polls_created }) .catch(e => { return { code: 500, data: e.originalStack }; }); </pre>	<pre> emoji, hash, friend_challenges, friend_challenges_won, tiki_score, polls_created }) .catch(e => { return { code: 500, data: e.originalStack }; }); if (!!result.code) { if (result.data.includes('UNIQUE constraint failed: users.username')) { return { code: 409, data: 'username already exists' } } return result; } </pre>
--	--

Unit Testing Defects / Testing Defect Log

Product	Settings		
Author	Cameron Glass		
Defect #	Description	Severity	How Corrected
1	Function resulted in error found when display name or password was updated	1	Separated methods for both each with their own verification functions.
2	Special characters were accepted in the display name.	3	Display name verification changed to check for special

			characters and other specified characters
3	Special characters were accepted in the password.	3	Password verification changed to check for special characters and other specified characters
4	Empty request sent to backend	1	Added pre-send checks to test to verify that variables needed were full
5	Invalid user in database/username doesn't exist	3	Proper error message was sent to the user
6	Password wasn't being securely hashed before being put in the database upon a password change	1	Added extra password hashing step

Product	Login		
Author	Nirali Rai		
Defect #	Description	Severity	How Corrected
1	Username entered was not checked for generic parameter checks	1	Added username verification statements to ensure username is in proper form
2	Password entered was not checked for generic parameter checks	2	Added password verification statements to ensure username is in proper form
3	Special characters were sent in request	2	Added checks to pre-sending the request
4	Empty request was sent	2	Added test checking for proper entries and ensuing needed parameters were present.
5	Username doesn't exist in the database	2	Added precheck which throws the correct HTTP status code and error

	which broke the request		data
6	Password is incorrect which broke the request	2	Added precheck which throws the correct HTTP status code and error data

Product	Sign up		
Author	Todd Griffin		
Defect #	Description	Severity	How Corrected
1	Invalid usernames were accepted	2	To allow for easy transition and information retrieval, usernames were standardized using regex
2	Displayname entry accepted any format of a display	2	For standardization , regexes were added to display name verifications
3	Password entry accepted any format password	2	For security reasons, more criteria were added to password format
4	Email entry accepted all inputs	2	Regex added that checked for standard email format
5	Password wasn't being securely hashed before being put in the database upon a password change	1	Added extra password hashing step
6	Username already exists in the database which broke the request	2	Added precheck which throws the correct HTTP status code and error data

Product	Forgot password
Author	Rishabh Ramsisaria

Defect #	Description	Severity	How Corrected
1	Account was not checked to make sure that it exists before allowing forgot password request	1	Made sure to check all users to confirm that the user does in fact exist when sending request
2	Username entered was not checked for generic parameter checks	1	Added username verification statements to ensure username is in proper form
3	Email entered was not checked for generic parameter checks	3	Added email verification statements to ensure username is in proper form
4	Special characters were sent in request	3	Added checks to pre-sending the request
5	Empty request was sent	1	Added test checking for proper entries and ensuing needed parameters were present.

Product	Profile Page		
Author	Dominic Miller		
Defect #	Description	Severity	How Corrected
1	Win rate was not calculated correctly based on passed in data	2	Revisited formulas for calculating win rate
2	Tiki Tally was calculated incorrectly based on the data passed to it	2	Revisited formulas for calculating Tiki Tally
3	Polls created did not display the right number of polls that were created	2	Made sure the polls created were counted correctly
4	Username doesn't exists in the database which broke the request	2	Added precheck which throws the correct HTTP status code and error data

Product	Private		
Author	Nirali Rai		
Defect #	Description	Severity	How Corrected
1	Themes return themes that contain partial substring of the input in the private page	2	Verifications for full string was added when searching by themes.
2	Empty sort request was sent in private page	2	When sorting checks if sorting option was selected prior to sending request
3	Created poll does not appear on private page view	1	Before confirmation, added check to make sure poll was stored in the backend

Product	Public		
Author	Cameron Glass		
Defect #	Description	Severity	How Corrected
1	Themes return themes that contain partial substring of the input in the public page	2	Verifications for full string was added when searching by themes.
2	Empty sort request was sent in public page	2	When sorting checks if sorting option was selected prior to sending request
3	Created poll does not appear on public page view	1	Before confirmation, added check to make sure poll was stored in the backend

Product	Create poll(Challenge)		
Author	Todd Griffin		

Defect #	Description	Severity	How Corrected
1	Title and theme were not required for poll to be created	2	Verification for submitting request was added. if missing, error message was sent to the user.
2	Sending poll to friend would not send a confirmation message if poll sent successfully	2	Statement was added to the frontend sending message if poll sent successfully.
3	Sending poll to friend would not send an error message if poll was not sent successfully	2	Statement was added to the frontend sending message if an error was returned.
4	Friends not selected when trying to send poll	1	Checked to make sure friend was selected before request sent, else error message displayed.

Product	Friends		
Author	Rishabh Ramsisaria		
Defect #	Description	Severity	How Corrected
1	Input for searching friend accepted any format	3	Added parameters and symbols to standardize
2	If friend not found no error messages were returned to the user	2	Statement was added to the frontend sending message if an error was returned.
3	If friend request was sent successfully, confirmation message was not sent	2	Message added in frontend to send a confirmation message