

Bouldering Buffs

Developers: Arielan Palencia Li, Kristin Off, Oliver Costello, Brie Brocato, and Suhani Agarwal

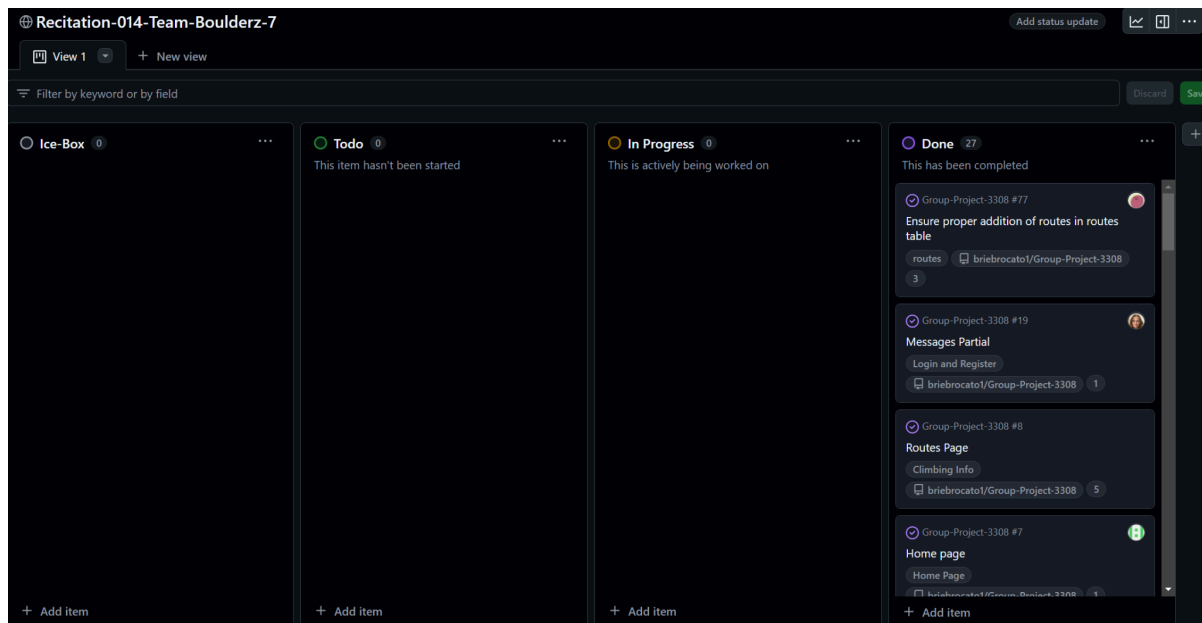
Project Description

Our project is a website designed to connect and support climbers in the Boulder area. The platform helps climbers find and share information about climbing routes through an easy-to-use database. Users can search for routes based on different criteria, view detailed descriptions, and leave ratings and reviews to help others choose their next climb.

The website also features a message board where climbers can plan and organize meetups. Users can post about upcoming climbs, invite others to join, and share details about time and location. This makes it simple to connect with fellow climbers and arrange group outings. By combining route discovery with a space to plan meetups, the site encourages connection and community. Whether you're new to climbing or an experienced climber, this platform offers tools to share knowledge, plan adventures, and build relationships within the Boulder climbing community.

Project Tracker

GitHub project board: <https://github.com/users/briebrocato1/projects/3>



Video Demonstration

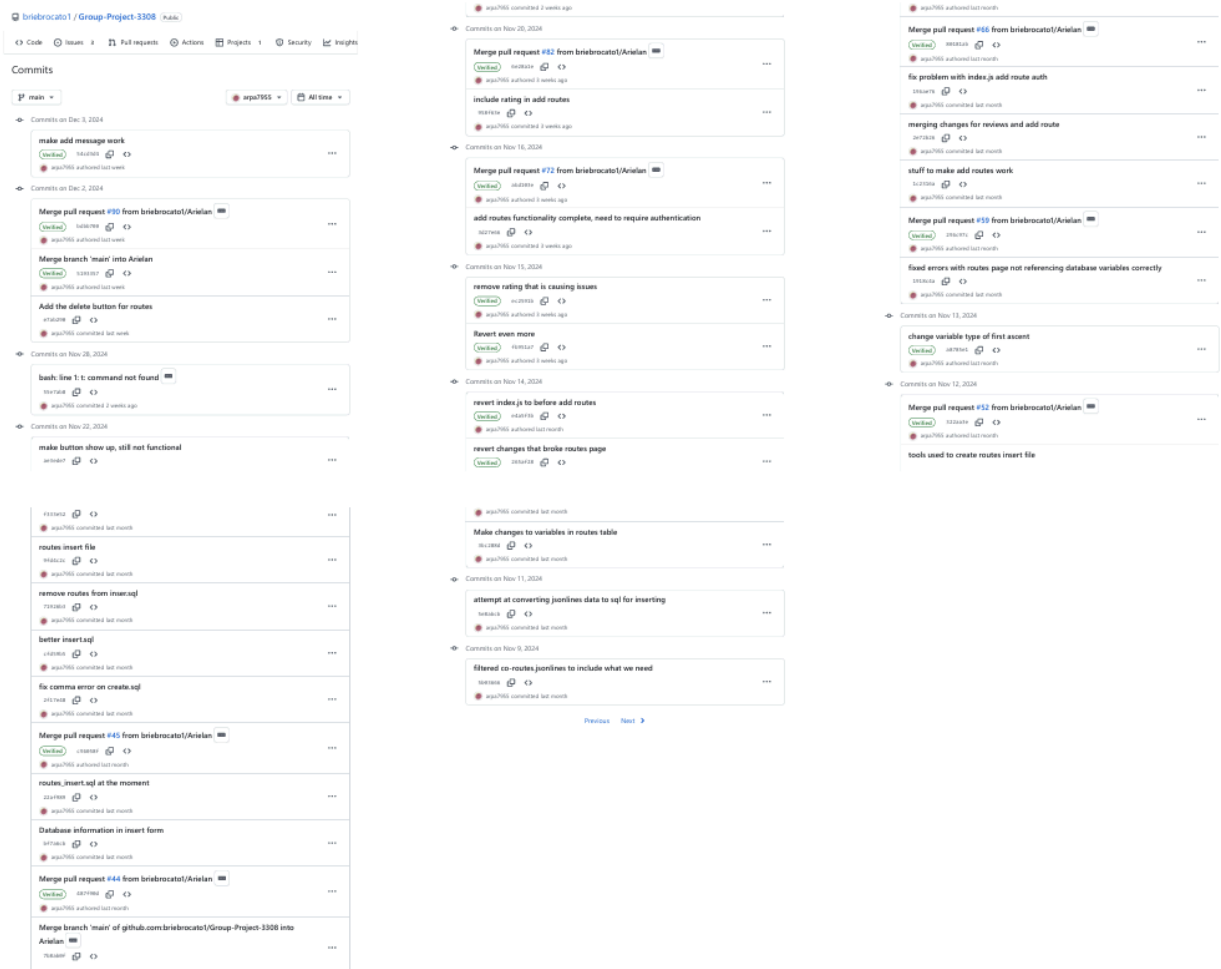
<https://drive.google.com/file/d/1oaRdpN8eRUE8Ktf6h6D1gUUkFPvCq2KE/view?usp=sharing>

Version Control System

<https://github.com/briebrocato1/Group-Project-3308>


























Developer Contributions

1. Arielan



My main focus for this project was creating a working routes page that displayed the routes from the climbing route information we got from a 2020 repository of the Mountain Project's information. This process included converting the data from a json lines file to sql statements, which required a Python script, JQuery statements, and manual deletion of problematic data. For the site itself I created the add routes option for logged in users, and a delete route option available only to admin, using HTML, Handlebars, and NodeJS.

2. Kristin

<p>Merge pull request #74 from briarocraft/Kristin </p> <p> briarocraft 4/28/2024  3 weeks ago</p>	
<p><i>Removed some unnecessary/unused code.</i></p> <p>4/28/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p>Merge pull request #75 from briarocraft/Kristin </p> <p> briarocraft 5/1/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p><i>Put display for route types back into route page.</i></p> <p>5/1/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p><i>Added login authentication to add routes.</i></p> <p>5/1/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p>Merge pull request #77 from briarocraft/Kristin </p> <p> briarocraft 5/1/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p><i>Minor change to style rules to make sure that modal buttons can be clicked for messages.</i></p> <p>5/1/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p>← Contents on Nov 15, 2024</p>	
<p>Merge pull request #70 from briarocraft/Kristin </p> <p> briarocraft 5/17/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	
<p><i>Fixed routes page to display properly with all recently added features.</i></p> <p>5/17/2024  3 weeks ago</p> <p> briarocraft commented 3 weeks ago</p>	

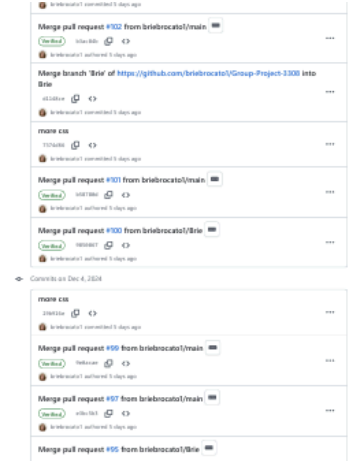
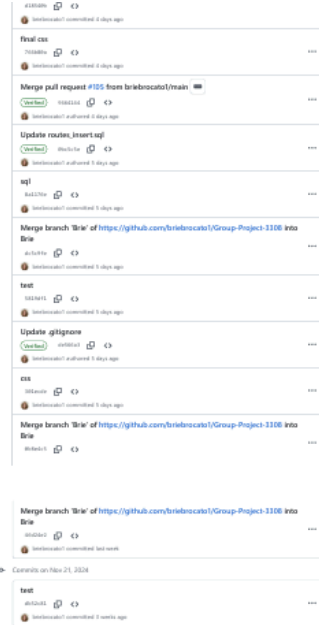
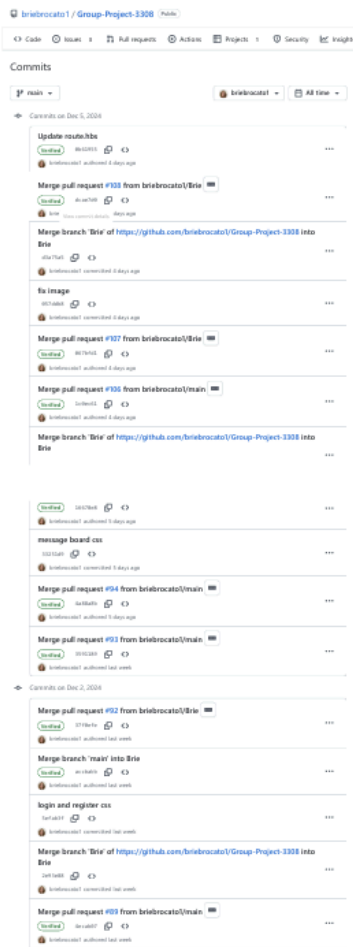
3. Ollie

<div> briebracato1 / Group-Project-3308 </div> <div> CodeIssuesPull requestsActionsProjectsSecurityInsights </div> <div>Commits</div> <div> main All time </div> <div> <div> Merge pull request #10 from briebracato/testing </div> <div> Merge branch 'main' into testing </div> </div> <div> <div> Update README.md </div> <div> Update README.md </div> <div> cleared old code from script.js </div> <div> Update insert.sql </div> </div> <div> <div> Create KickStart.txt </div> <div> Update co-areas.jsonlines </div> <div> Update co-routes.jsonlines </div> <div> Add files via upload </div> <div> Update README.md </div> </div>	<div> Merge pull request #87 from briebracato/testing </div> <div> Merge pull request #83 from briebracato/individualroute </div> <div> Merge pull request #75 from briebracato/routefilter </div> <div> Merge branch 'main' into routefilter </div> <div> Merge pull request #69 from briebracato/routefilter </div> <div> Merge branch 'main' into routefilter </div> <div> Merge pull request #51 from briebracato/routespage </div> <div> Merge pull request #50 from briebracato/routespage </div> <div> Create server.spec.js </div> <div> Create insert.sql </div> <div> Create create.sql </div> <div> Create index.js </div>
---	---

PreviousNext

Most of my time was spent implementing the message board and the search/filter on the routes page, as well as a few other simpler pages, such as the home page or single route page. I also worked on the user acceptance tests, the SQL database, and various bug fixes. I worked on our main javascript routes, on the nodeJS, on HTML and Handlebars, and on the Mocha and Chai tests.

4. Brie



My contributions to the group project focused on improving the visual and functional appeal of our website. I extensively worked on the CSS of our project to improve our website design and user interface. I contributed to a few debugging issues as well, such as SQL-related problems, errors with our database, and issues with our routes to pages. I also collaborated with my group on the index.js file to handle our routing. I contributed to smaller pages like the footer and changed our header to properly handle the css for our pages.

5. Suhani

Message Board Page

Nav Bar

Text w/ Page Title

Image of Mountains

Add Button

Text w/ Message

Reply Delete

Text w/ Message

Reply Delete

Text w/ Message

Reply Delete

Individual Route Page

Nav Bar

Route Name

Route Info

Add Review

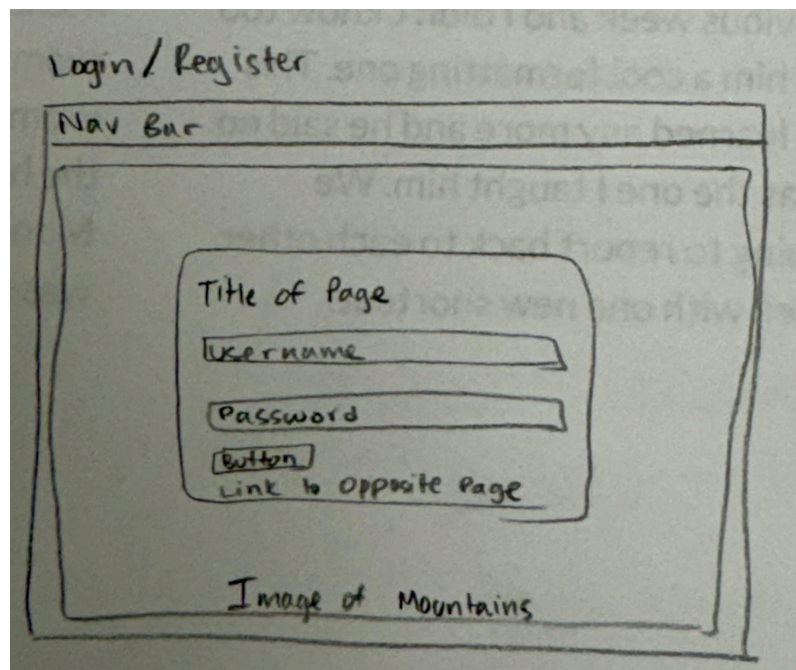
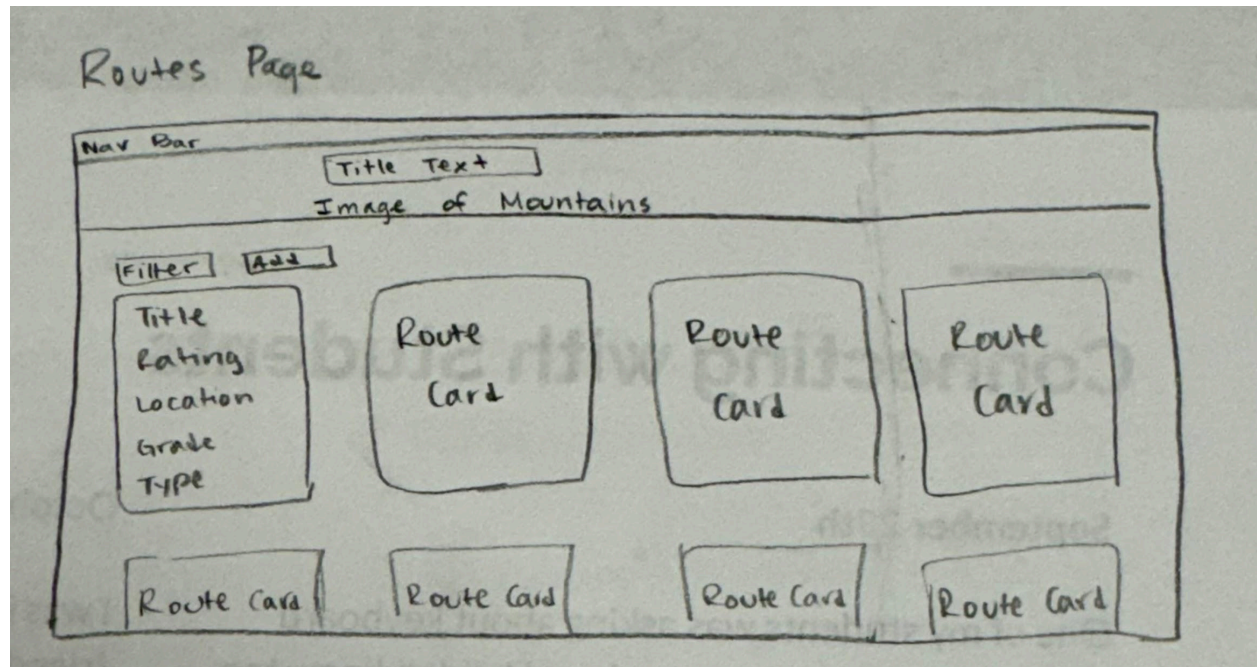
Name

Rating

Review

Submit

Image of Mountains



Test results and Observations

```

web-1 Database connection successful
web-1   ✓ Returns the default welcome message (93ms)
web-1
web-1 Testing register API
web-1   ✓ positive : /register (965ms)
web-1   ✓ Negative : /register. Checking invalid name
web-1
web-1 Testing login API
web-1 Session saved: {
web-1   username: 'olco',
web-1   email: 'olco2433@colorado.edu',
web-1   password: '$2a$10$Q/dqqYBSCI0yBRDvrFFeduJQxCa@aqcVJJeofRmuWNWoIR/eEMyCChC'
web-1 }
web-1   ✓ positive : /login (201ms)
db-1 2024-12-10 02:47:50.972 UTC [77] ERROR: operator does not exist: character varying = integer at character 36
db-1 2024-12-10 02:47:50.972 UTC [77] HINT: No operator matches the given name and argument types. You might need to add explicit type casts.
db-1 2024-12-10 02:47:50.972 UTC [77] STATEMENT: SELECT * FROM users WHERE username = 10;
web-1 error: operator does not exist: character varying = integer
web-1   at Parser.parseErrorMessage (/repository/node_modules/pg-protocol/dist/parser.js:283:98)
web-1   at Parser.handlePacket (/repository/node_modules/pg-protocol/dist/parser.js:122:29)
web-1   at Parser.parse (/repository/node_modules/pg-protocol/dist/parser.js:35:38)
web-1   at Socket.<anonymous> (/repository/node_modules/pg-protocol/dist/index.js:11:42)
web-1   at Socket.emit (node:events:519:28)
web-1   at addChunk (node:internal/streams/readable:559:12)
web-1   at readableAddChunkPushByteMode (node:internal/streams/readable:510:3)
web-1   at Readable.push (node:internal/streams/readable:390:5)
web-1   at TCP.onStreamRead (node:internal/stream_base_commons:191:23) {
web-1   length: 212,
web-1   severity: 'ERROR',
web-1   code: '42883',
web-1   detail: undefined,
web-1   hint: 'No operator matches the given name and argument types. You might need to add explicit type casts.',
web-1   position: '36',
web-1   internalPosition: undefined,
web-1   internalQuery: undefined,
web-1   where: undefined,
web-1   schema: undefined,
web-1   table: undefined,
web-1   column: undefined,
web-1   dataType: undefined,
web-1   constraint: undefined,
web-1   file: 'parse_oper.c',
web-1   line: '647',
web-1   routine: 'op_error',
web-1   query: 'SELECT * FROM users WHERE username = 10;',
web-1   params: undefined
web-1 }
web-1   ✓ Negative : /login. Checking invalid name (61ms)

```

```

web-1   ✓ Negative : /login. Checking invalid name (61ms)
web-1
web-1 Positive: Testing Render
web-1   ✓ test "/home" route should render with an html response (58ms)
web-1
web-1 Negative: Testing Redirect
web-1   ✓ login route should redirect to /login (83ms)
web-1
web-1 Testing add route API
web-1 test V0 PG true true true true test1 test2 111.111 222.222 test3 tester 5
web-1   ✓ positive : /add-route (164ms)
web-1 undefined undefined undefined false false false false undefined undefined null null noName undefined undefined
db-1 2024-12-10 02:47:51.339 UTC [77] ERROR: null value in column "routename" of relation "routes" violates not-null constraint
db-1 2024-12-10 02:47:51.339 UTC [77] DETAIL: Failing row contains (633, null, null, null, f, f, f, f, f, null, null, null, null, noName, null, null, 0, f).
web-1 Error adding route: error: null value in column "routename" of relation "routes" violates not-null constraint
db-1 2024-12-10 02:47:51.339 UTC [77] STATEMENT: INSERT INTO routes (routeName, grade, safety, sport, trad, toprope, boulder, snow, alpine, description, location, areaLatitude, areaLongitude, areaName, firstAscent, rating)
web-1   at Parser.parseErrorMessage (/repository/node_modules/pg-protocol/dist/parser.js:283:98)
db-1   at Parser.handlePacket (/repository/node_modules/pg-protocol/dist/parser.js:122:29)
web-1   at Parser.parse (/repository/node_modules/pg-protocol/dist/parser.js:35:38)
web-1   at Socket.<anonymous> (/repository/node_modules/pg-protocol/dist/index.js:11:42)
web-1   at Socket.emit (node:events:519:28)
web-1   at addChunk (node:internal/streams/readable:559:12)
web-1   at readableAddChunkPushByteMode (node:internal/streams/readable:510:3)
web-1   at Readable.push (node:internal/streams/readable:390:5)
web-1   at TCP.onStreamRead (node:internal/stream_base_commons:191:23) {
web-1   length: 287,
web-1   severity: 'ERROR',
web-1   code: '23502',
web-1   detail: 'Failing row contains (633, null, null, null, f, f, f, f, f, null, null, null, null, noName, null, null, 0, f).',
web-1   hint: undefined,
web-1   position: undefined,
web-1   internalPosition: undefined,
web-1   internalQuery: undefined,
web-1   where: undefined,
web-1   schema: 'public',
web-1   table: 'routes',
web-1   column: 'routename',
web-1   dataType: undefined,
web-1   constraint: undefined,
web-1   file: 'execMain.c',
web-1   line: '1883',
web-1   routine: 'ExecConstraints',
web-1   query: 'INSERT INTO routes (routeName, grade, safety, sport, trad, toprope, boulder, snow, alpine, description, location, areaLatitude, areaLongitude, areaName, firstAscent, rating)\n' +
web-1     'VALUES (null, null, null, false, false, false, false, false, null, null, null, null, null, noName, null, null, 0, f)',
web-1   params: undefined
web-1 }
web-1   ✓ Negative : /add-route. Missing fields
web-1 test V0 PG true true true true test1 test2 111.111 222.222 test3 tester 5
web-1   ✓ Negative : /add-route. Duplicate
web-1
web-1 testing Routes API
web-1   ✓ /routes should display HTML
web-1   ✓ /routes should return routes filtered by name (109ms)
web-1   ✓ /routes should display "no routes available" when no routes match the query (62ms)
web-1   ✓ /routes should display all routes when no query parameters are provided (101ms)
web-1
web-1 14 passing (2s)

```

Test Set 1: Users will be prompted to register an account if they do not already have an account when they log in. If the user attempts to log in, and their credentials are not found in our accounts table, they are redirected to the registration page.

- Test Results
 - Pass Criteria: Each test case should complete with the correct expected outcome. For example, if a user logs in with an account that doesn't exist, they should be redirected to the registration page.
 - Fail Criteria: Any deviations from the expected outcomes will be marked as failures and recorded with error logs and screenshots to be reported to the development team.
- Observations
 - What are the users doing?
 - Users are logging in or registering accounts based on their credentials.
 - What is the user's reasoning for their actions?
 - Users aim to access the application by logging in or creating an account if they don't have one.
 - Is their behavior consistent with the use case?
 - Yes, users follow the expected flow: logging in successfully (Test 1), being redirected to registration if no account exists (Test 2 and 3), and receiving error messages for duplicate registrations (Test 4).
 - If there is a deviation from the expected actions, what is the reason for that?
 - No major deviations were observed; possible issues might involve input errors or bugs in redirection.
 - Did you use that to make changes to your application?
 - Yes, improvements were made:
 - Enhanced error messages for clarity.
 - Added visual feedback during login/registration.
 - Strengthened input validation to prevent errors.
 - These changes ensure smoother user interactions and alignment with expectations.

Test Set 2: The feature allows users to filter climbing routes based on the route's location. This test ensures that the filtering functionality works correctly, providing users with relevant results when searching for climbing routes in specified locations.

- Test Results
 - Pass Criteria: Each test case should yield the expected filtered results based on the chosen location, error messages for invalid inputs, and no unexpected results for non-existent locations.

- Fail Criteria: Any result that does not align with the expected outcomes will be documented, with relevant error logs, screenshots, and steps to reproduce, to be shared with the development team.
- Observations
 - What are the users doing?
 - Users are filtering climbing routes by selecting specific locations, testing the system's ability to return accurate and relevant results.
 - What is the user's reasoning for their actions?
 - Users aim to narrow down climbing route options to a specific area, improving the relevance and efficiency of their search experience.
 - Is their behavior consistent with the use case?
 - Yes, users followed expected behaviors: selecting a valid location (Test 1), encountering a message for unavailable or invalid locations (Test 2), and clearing filters to view all routes (Test 3).
 - If there is a deviation from the expected actions, what is the reason for that?
 - No significant deviations were observed. Potential deviations could arise from database inaccuracies, input errors, or lack of inputs.
 - Did you use that to make changes to your application?
 - Yes, updates included:
 - Improved messaging for invalid locations to make errors clearer.
 - Enhanced UI feedback when filters are applied.
 - Requiring specific data types for inputs.
 - These refinements ensure the feature meets user needs and provides a seamless experience.

Test Set 3: This feature allows users to add new climbing routes by filling out a form with relevant information (e.g., route name, location, difficulty, description). This test ensures that the form submission functionality works correctly, validates inputs, and saves new climbing routes to the routes page.

- Test Results
 - Pass Criteria: Each test case should lead to the correct response for valid and invalid inputs. Routes added with valid data should display accurately on the routes page, while errors should prompt users to correct issues with invalid data entries.
 - Fail Criteria: Any unexpected result, such as improper data saving or missing error messages, will be documented with error logs, screenshots, and reproduction steps to be communicated to the development team.
- Observations
 - What are the users doing?

- Users are adding new climbing routes by filling out a form with route details and submitting it.
- What is the user's reasoning for their actions?
 - Users want to contribute new climbing route information to the database, making it accessible for future searches and exploration.
- Is their behavior consistent with the use case?
 - Yes, users are completing the form as expected: successfully adding new routes (Test 1), encountering error messages for missing fields (Test 2), and receiving warnings for duplicate submissions (Test 3).
- If there is a deviation from the expected actions, what is the reason for that?
 - No major deviations were noted. Minor issues, such as users omitting fields unintentionally or attempting duplicate entries, were anticipated and handled appropriately.
- Did you use that to make changes to your application?
 - Yes, adjustments included:
 - Enhanced error messages to specify which required fields are missing.
 - Stopped users from being able to submit routes when missing required fields
 - Improved duplicate detection to check for similar routes more robustly.
 - These changes enhance user experience and ensure the feature works as intended.

Deployment

<https://bouldering-buffs.onrender.com/>