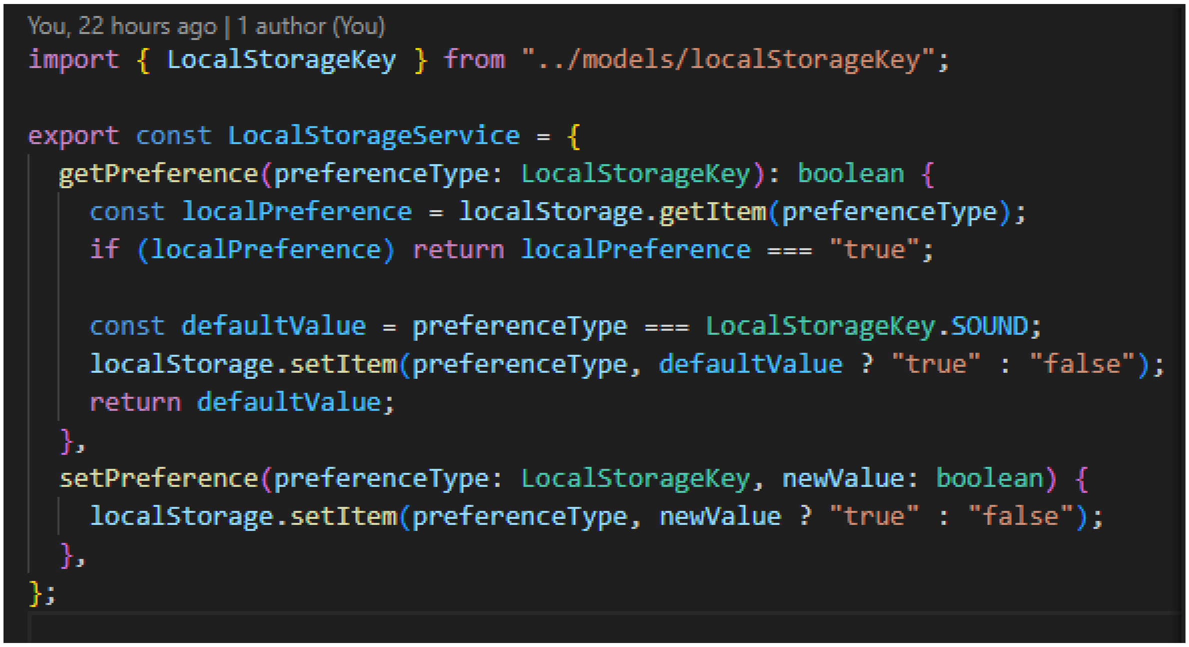
Benson Bird
Advanced Front End
12/9/2024

Project scope is 2-3 times larger than Inventory Management (per group member)

I would say that I got around 75%, not more than that for this criteria. I don't know how to show you that, but it is very big and thought out. Sadly it is not totally the giant game I wanted it to be.

Technology: use local storage

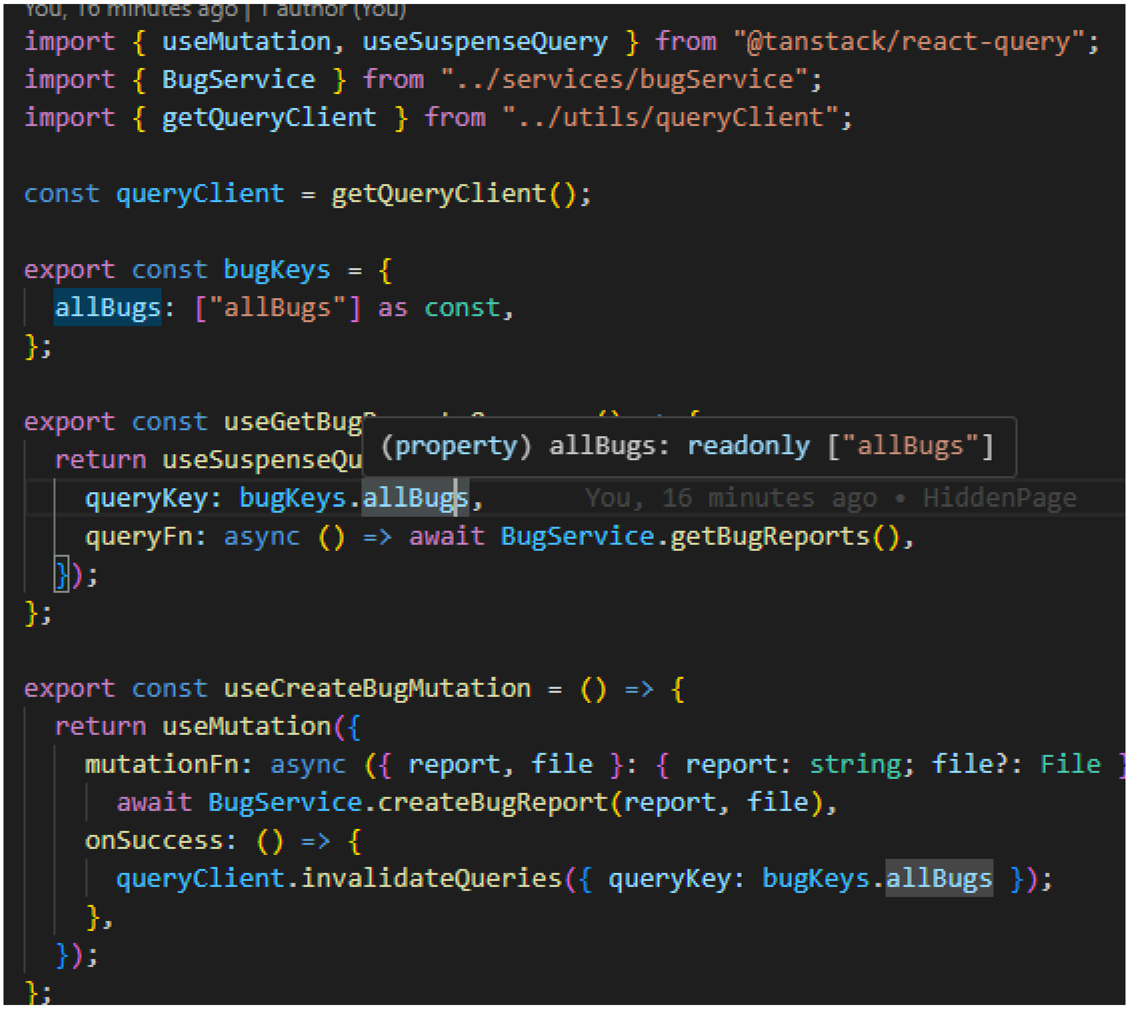
I use it to store the user's preference for notifications and sound. Different devices will probably want different preferences.



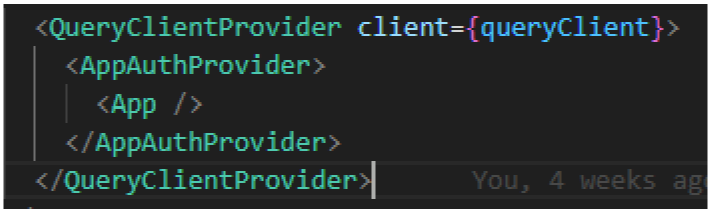
You, 22 hours ago | 1 author (You)

```
import { LocalStorageKey } from "../models/localStorageKey";

export const LocalStorageService = {
  getPreference(preferenceType: LocalStorageKey): boolean {
    const localPreference = localStorage.getItem(preferenceType);
    if (localPreference) return localPreference === "true";

    const defaultValue = preferenceType === LocalStorageKey.SOUND;
    localStorage.setItem(preferenceType, defaultValue ? "true" : "false");
    return defaultValue;
  },
  setPreference(preferenceType: LocalStorageKey, newValue: boolean) {
    localStorage.setItem(preferenceType, newValue ? "true" : "false");
  },
};
```

Technology: Client side state stores (e.g. tanstack query or context)

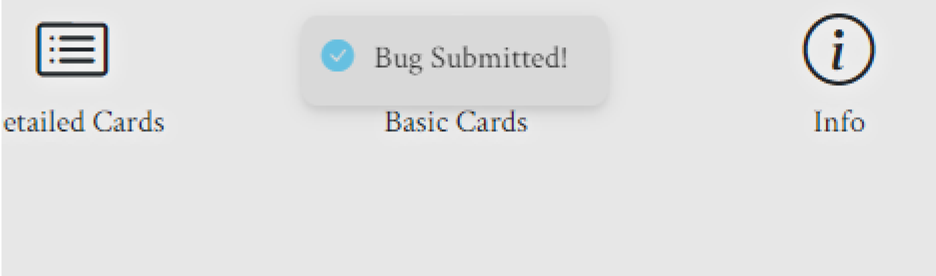


```
You, 16 minutes ago | 1 author (You)
import { useMutation, useSuspenseQuery } from "@tanstack/react-query";
import { BugService } from "../services/bugService";
import { getQueryClient } from "../utils/queryClient";

const queryClient = getQueryClient();

export const bugKeys = {
  allBugs: ["allBugs"] as const,
};

export const useGetBug
  return useSuspenseQu  (property) allBugs: readonly ["allBugs"]
    queryKey: bugKeys.allBugs,        You, 16 minutes ago • HiddenPage
    queryFn: async () => await BugService.getBugReports(),
  });
};

export const useCreateBugMutation = () => {
  return useMutation({
    mutationFn: async ({ report, file }: { report: string; file?: File }
      await BugService.createBugReport(report, file),
    onSuccess: () => {
      queryClient.invalidateQueries({ queryKey: bugKeys.allBugs });
    },
  });
};
```

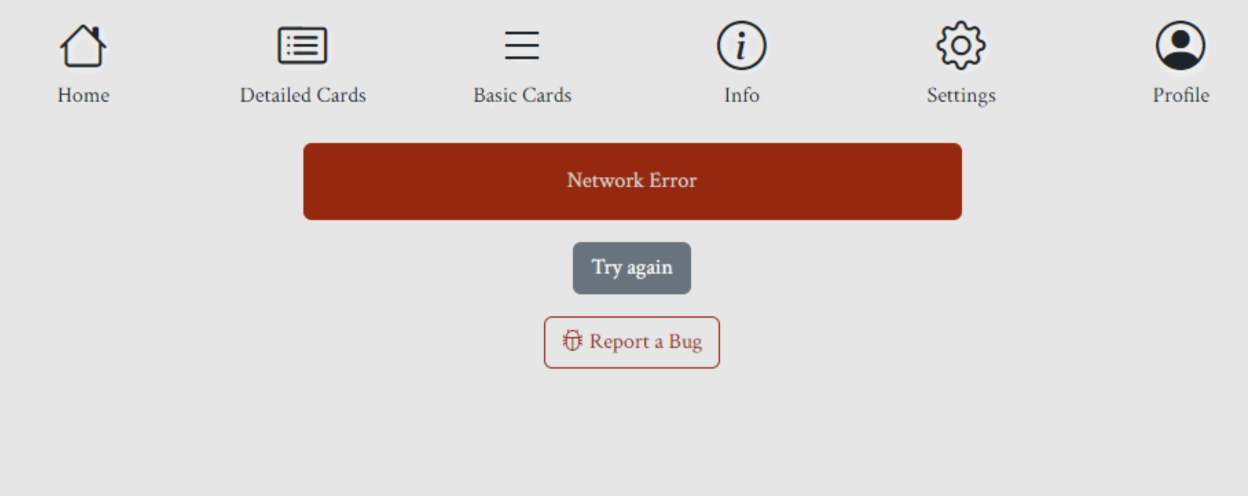


```
<QueryClientProvider client={queryClient}>
  <AppAuthProvider>
    <App />
  </AppAuthProvider>
</QueryClientProvider>       You, 4 weeks ag
```

Technology: Toasts / global notifications or alerts



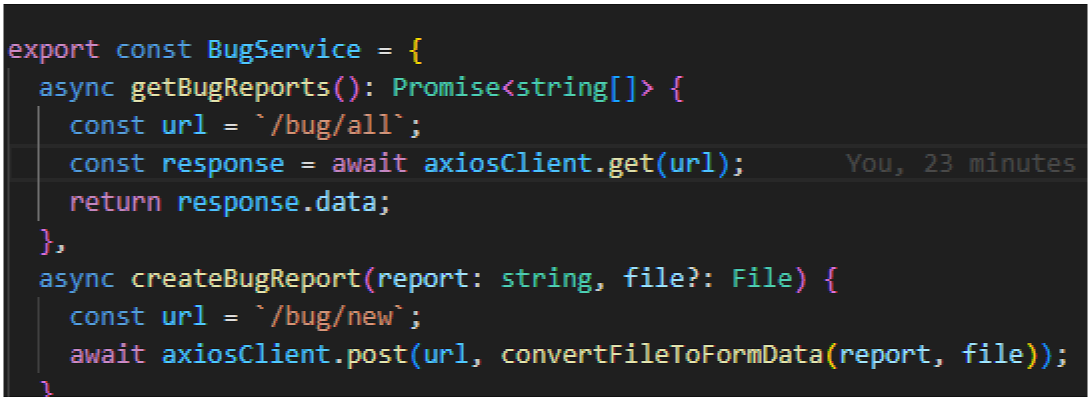
Technology: Error handling



Network Error

I didn’t do much on the backend of error handling.

Technology: Network Calls that read and write data

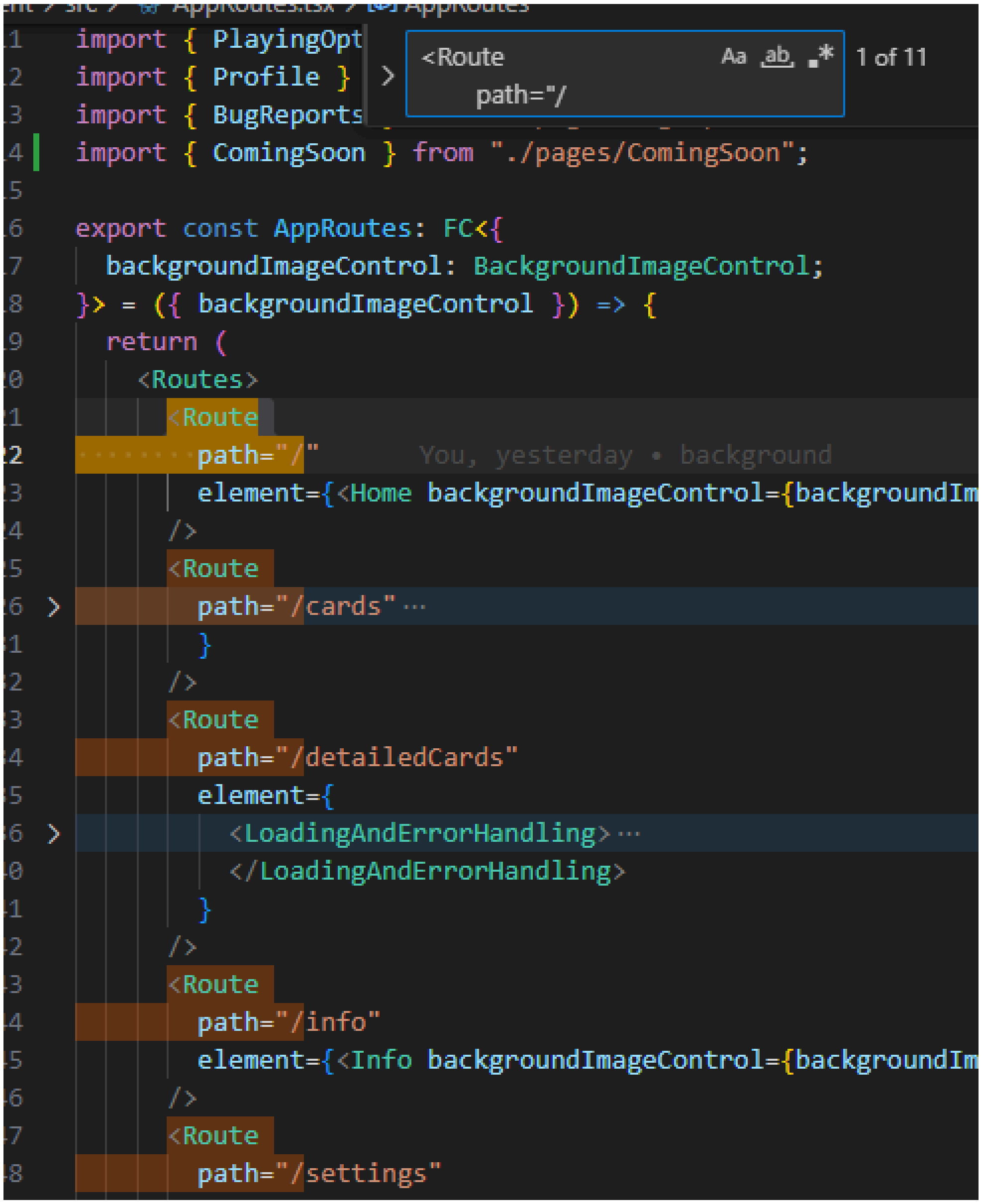


```
export const BugService = {
  async getBugReports(): Promise<string[]> {
    const url = `/bug/all`;
    const response = await axiosClient.get(url);        You, 23 minutes
    return response.data;
  },
  async createBugReport(report: string, file?: File) {
    const url = `/bug/new`;
    await axiosClient.post(url, convertFileToFormData(report, file));
  }
```

Technology: Developer type helping (typescript)

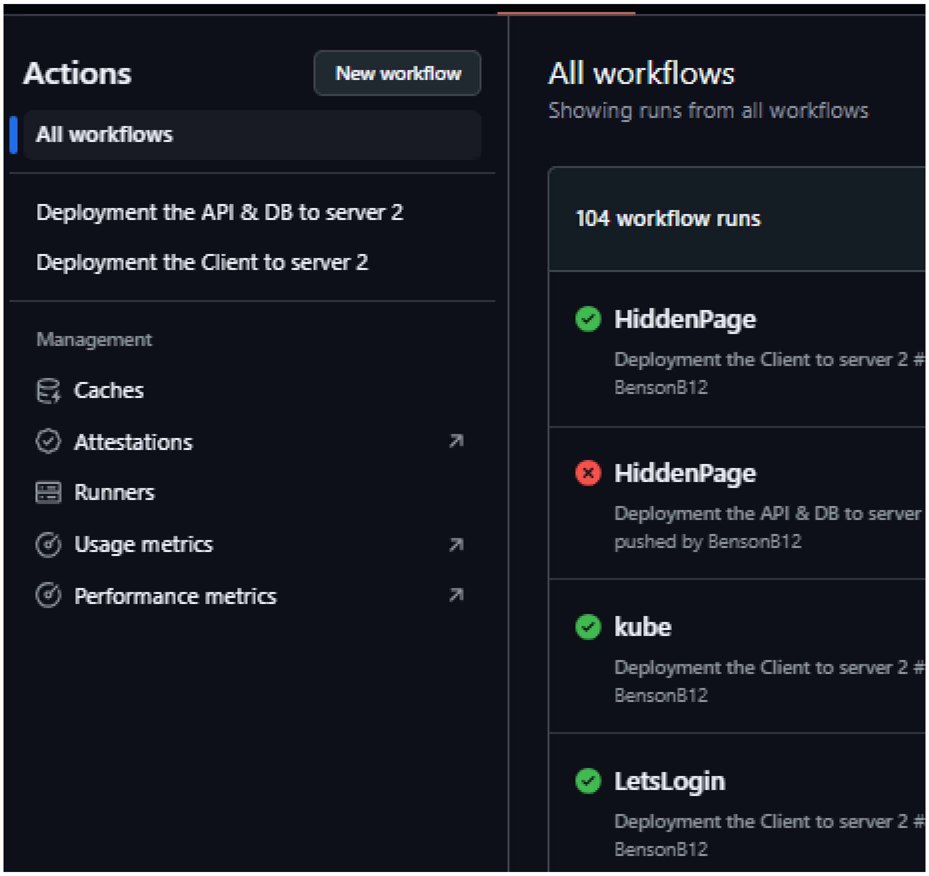
YES! See any coding picture above or below

Technology: 10+ pages or views
Despite the Ctrl + F - I only actually have about 8 pages. Give or take.


nt > src > AppRoutes.tsx > AppRoutes
<Route
path="/
Aa ab .*
1 of 11
You, yesterday • background

```
import { PlayingOpt
import { Profile }
import { BugReports
import { ComingSoon } from "./pages/ComingSoon";

export const AppRoutes: FC<{
  backgroundImageControl: BackgroundImageControl;
}> = ({ backgroundImageControl }) => {
  return (
    <Routes>
      <Route
        path="/"
        element={<Home backgroundImageControl={backgroundIm
      />
      <Route
        path="/cards" ···
        }
      />
      <Route
        path="/detailedCards"
        element={
          <LoadingAndErrorHandling> ···
          </LoadingAndErrorHandling>
        }
      />
      <Route
        path="/info"
        element={<Info backgroundImageControl={backgroundIm
      />
      <Route
        path="/settings"
```

CI/CD pipeline



All workflows

Showing runs from all workflows

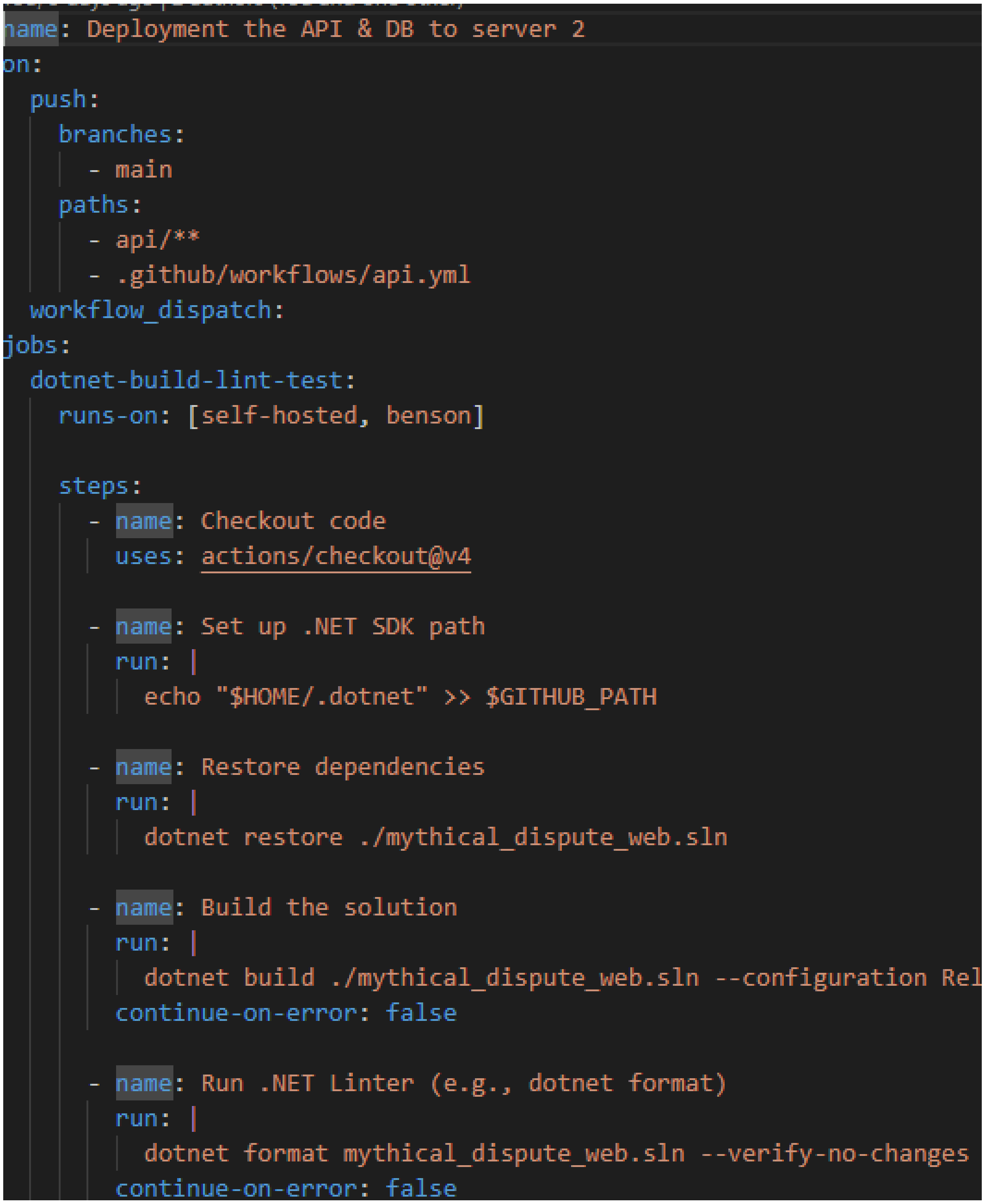
104 workflow runs

✓ **HiddenPage**
Deployment the Client to server 2 #
BensonB12

✗ **HiddenPage**
Deployment the API & DB to server
pushed by BensonB12

✓ **kube**
Deployment the Client to server 2 #
BensonB12

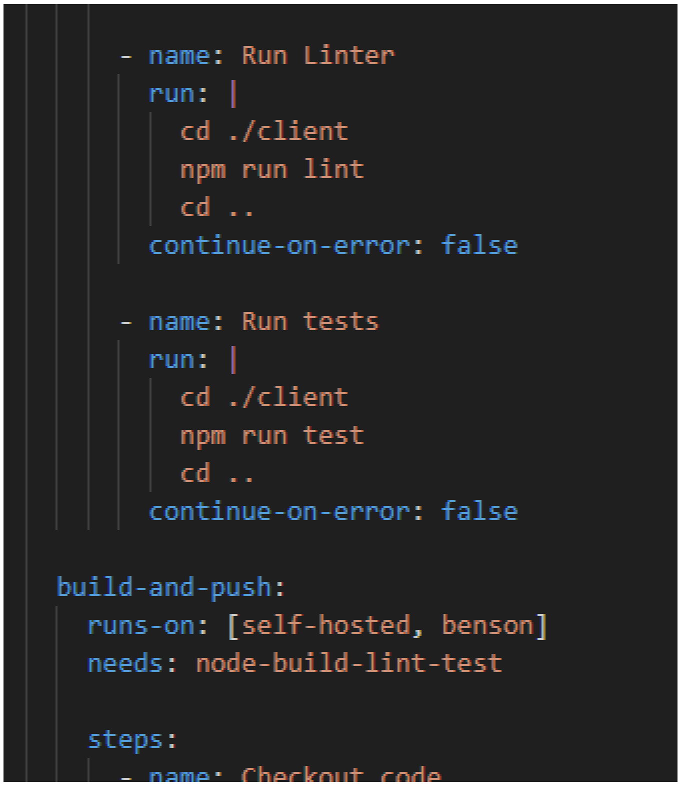
✓ **LetsLogin**
Deployment the Client to server 2 #
BensonB12



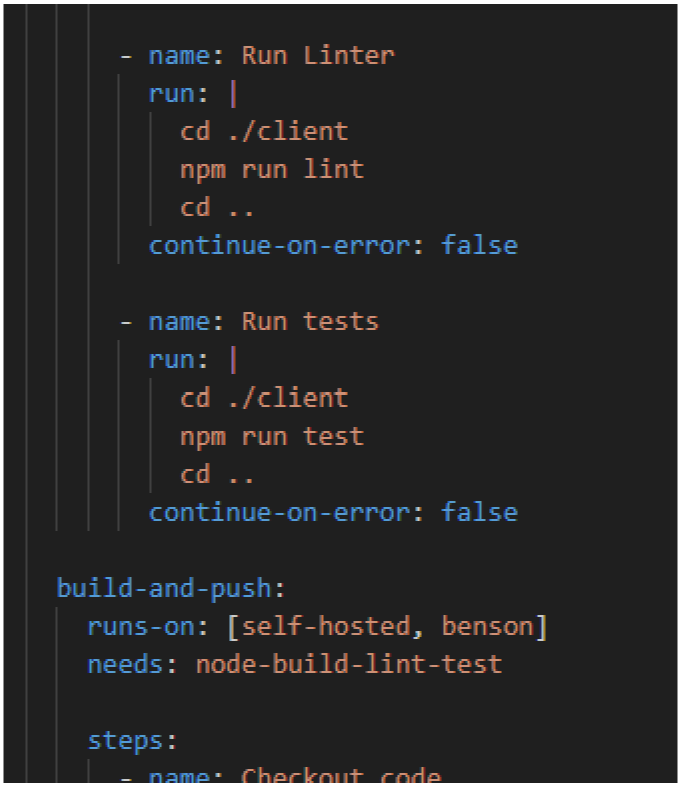
```yaml
name: Deployment the API & DB to server 2
on:
  push:
    branches:
      - main
    paths:
      - api/**
      - .github/workflows/api.yml
  workflow_dispatch:
jobs:
  dotnet-build-lint-test:
    runs-on: [self-hosted, benson]

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Set up .NET SDK path
        run: |
          echo "$HOME/.dotnet" >> $GITHUB_PATH

      - name: Restore dependencies
        run: |
          dotnet restore ./mythical_dispute_web.sln

      - name: Build the solution
        run: |
          dotnet build ./mythical_dispute_web.sln --configuration Rel
        continue-on-error: false

      - name: Run .NET Linter (e.g., dotnet format)
        run: |
          dotnet format mythical_dispute_web.sln --verify-no-changes
        continue-on-error: false
```

And more.

tests run in pipeline, pipeline aborts if they fail
&&
Technology: linting in pipeline

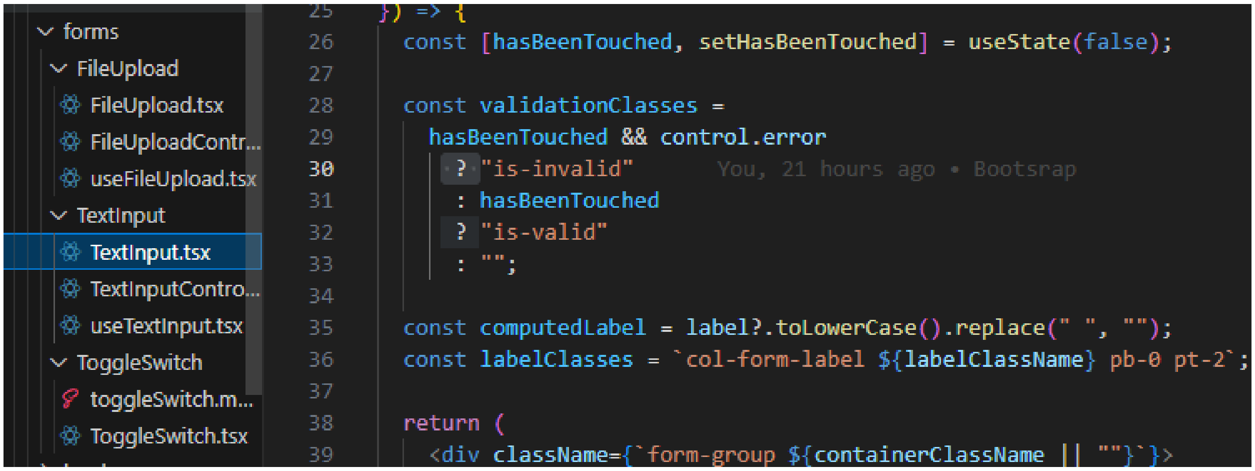


```
      - name: Run Linter
        run: |
          cd ./client
          npm run lint
          cd ..
        continue-on-error: false

      - name: Run tests
        run: |
          cd ./client
          npm run test
          cd ..
        continue-on-error: false

  build-and-push:
    runs-on: [self-hosted, benson]
    needs: node-build-lint-test

    steps:
      - name: Checkout code
```



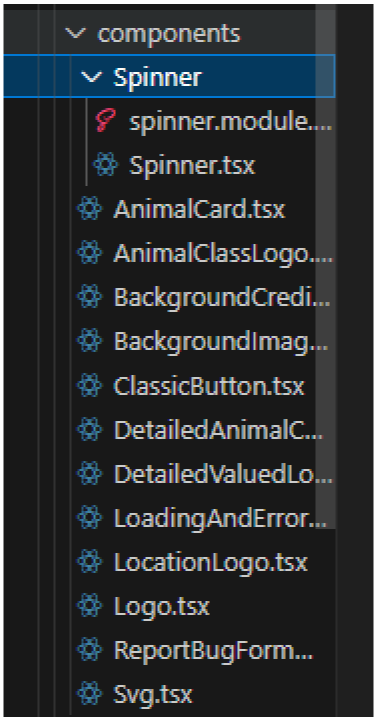
```
      - name: Run Linter
        run: |
          cd ./client
          npm run lint
          cd ..
        continue-on-error: false

      - name: Run tests
        run: |
          cd ./client
          npm run test
          cd ..
        continue-on-error: false

  build-and-push:
    runs-on: [self-hosted, benson]
    needs: node-build-lint-test

    steps:
      - name: Checkout code
```

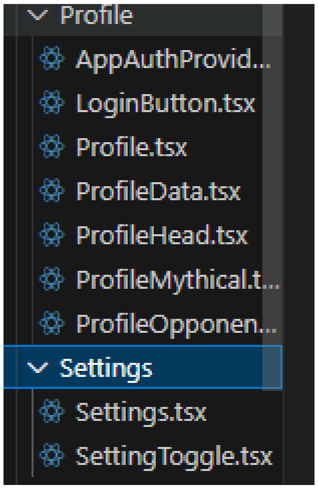
Technology: 3+ generic form input components



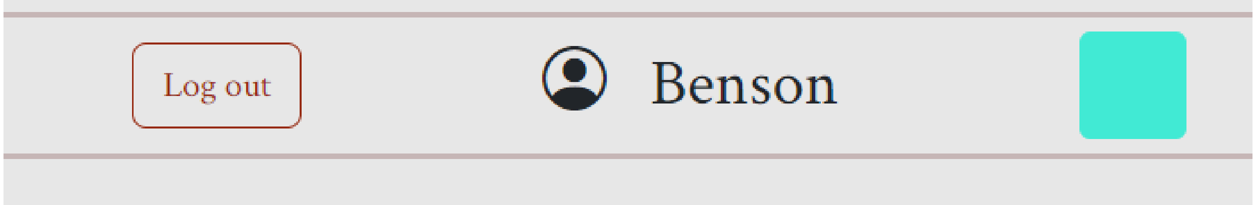
```
}) => {
  const [hasBeenTouched, setHasBeenTouched] = useState(false);

  const validationClasses =
    hasBeenTouched && control.error
      ? "is-invalid"       You, 21 hours ago • Bootsrap
      : hasBeenTouched
      ? "is-valid"
      : "";

  const computedLabel = label?.toLowerCase().replace(" ", "");
  const labelClasses = `col-form-label ${labelClassName} pb-0 pt-2`;

  return (
    <div className={`form-group ${containerClassName || ""}`}>
```

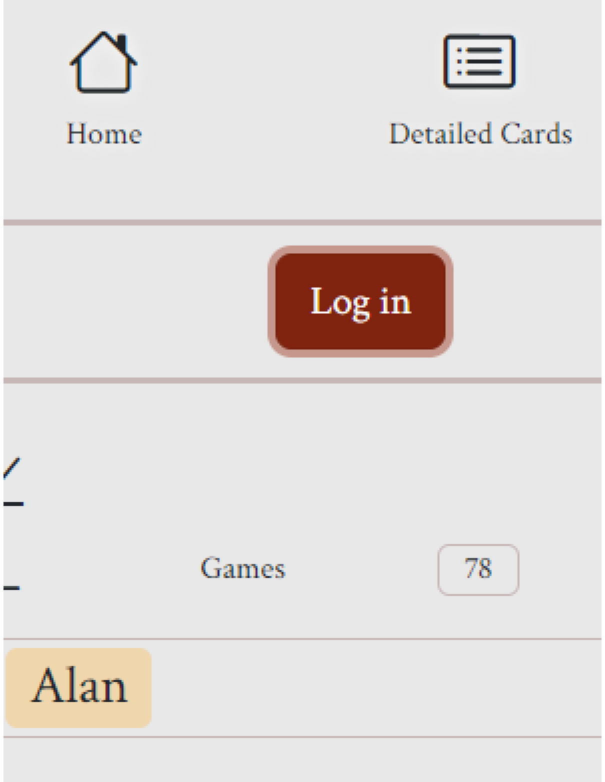
Technology: 4+ generic layout components





Technology: authentication and user account support

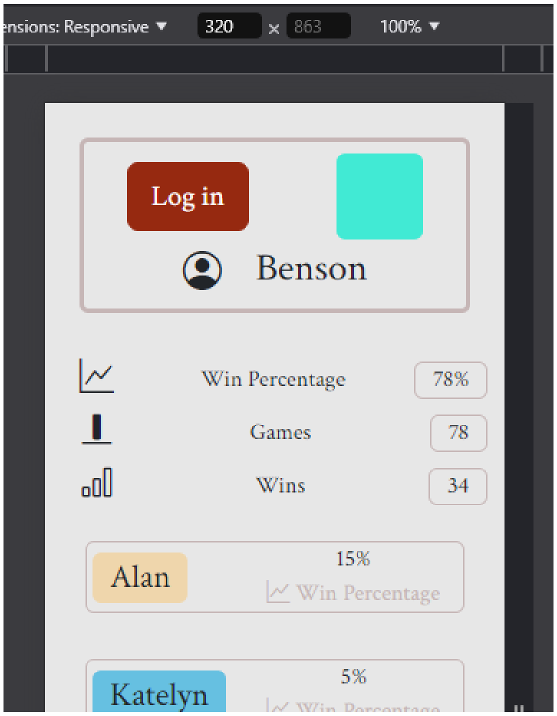




Technology: authorized pages and public pages

I tried and failed at this one. I call an api that handles if authorized worked, and show lost page if my hook throws 401. Not only is that not exactly what you are looking for, it also doesn't fully work. But I tried.

Experience: all experiences mobile friendly
Absolutely

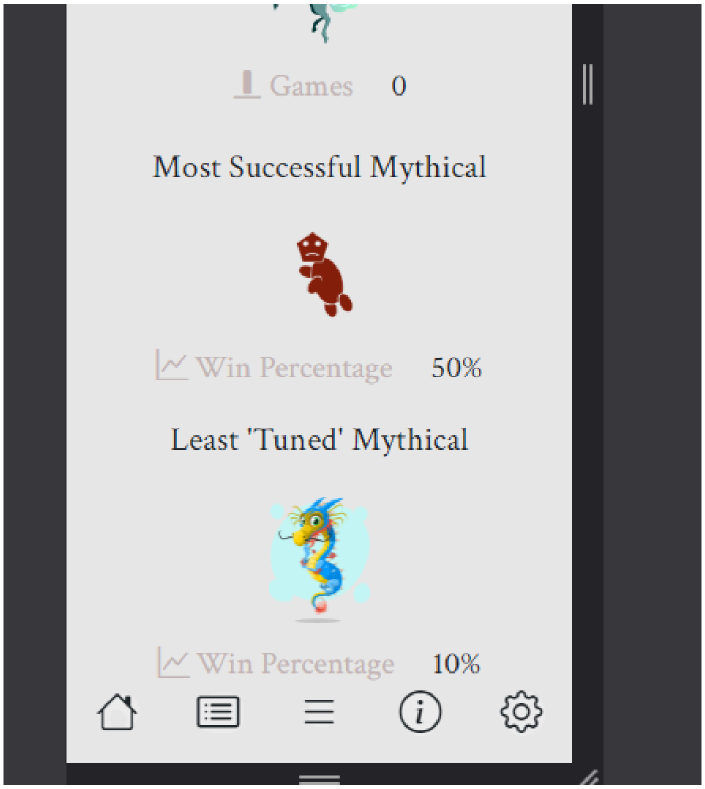


Benson

	Win Percentage	78%
	Games	78
	Wins	34

Alan 15% Win Percentage

Katelyn 5% Win Percentage



Games 0

Most Successful Mythical

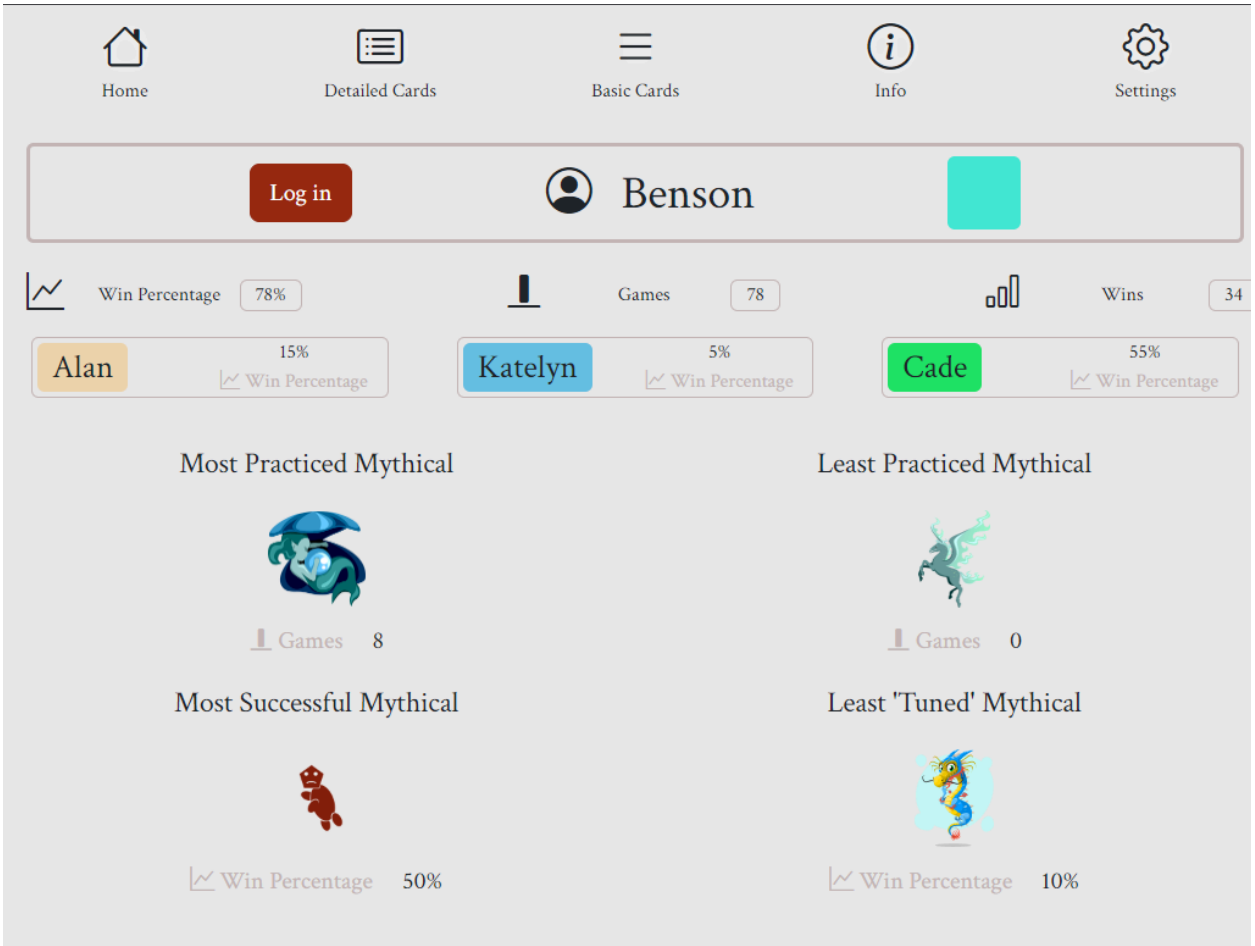
Win Percentage 50%

Least 'Tuned' Mythical

Win Percentage 10%

Experience: 3 instances where elements re-order themselves on smaller screens

See above for before, and after is below



Benson

Win Percentage 78%

Games 78

Wins 34

Alan 15% Win Percentage

Katelyn 5% Win Percentage

Cade 55% Win Percentage

Most Practiced Mythical

Games 8

Least Practiced Mythical

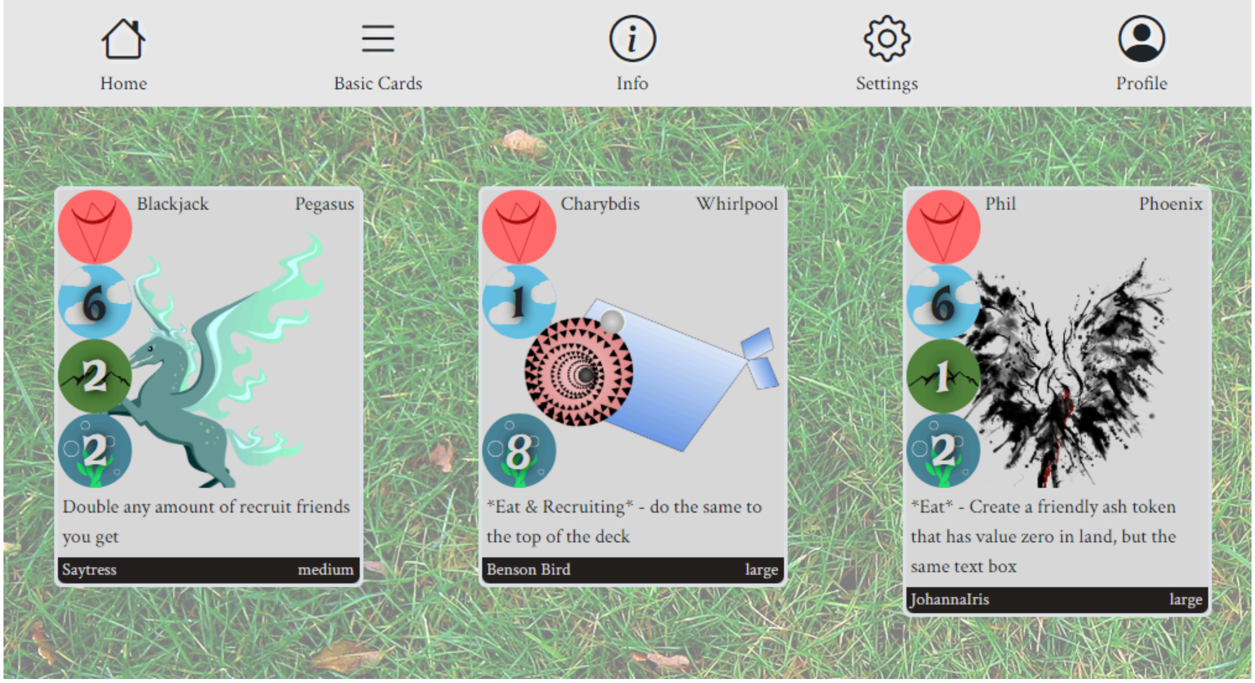
Games 0

Most Successful Mythical

Win Percentage 50%

Least 'Tuned' Mythical

Win Percentage 10%



Blackjack Pegasus

Double any amount of recruit friends you get

Saytress medium

Charybdis Whirlpool

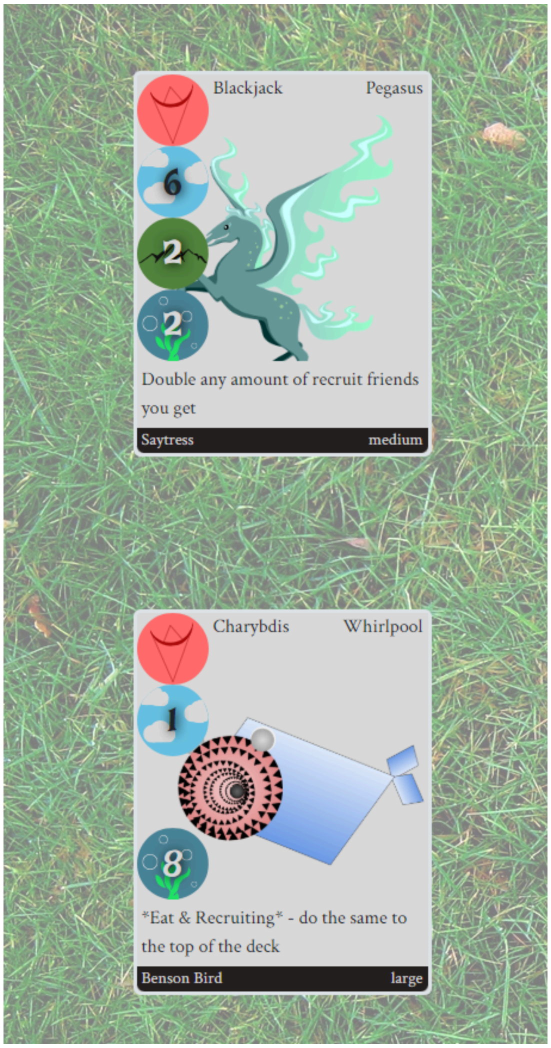
Eat & Recruiting - do the same to the top of the deck

Benson Bird large

Phil Phoenix

Eat - Create a friendly ash token that has value zero in land, but the same text box

JohannaIris large



Blackjack Pegasus

6

2

2

Double any amount of recruit friends you get

Saytress medium

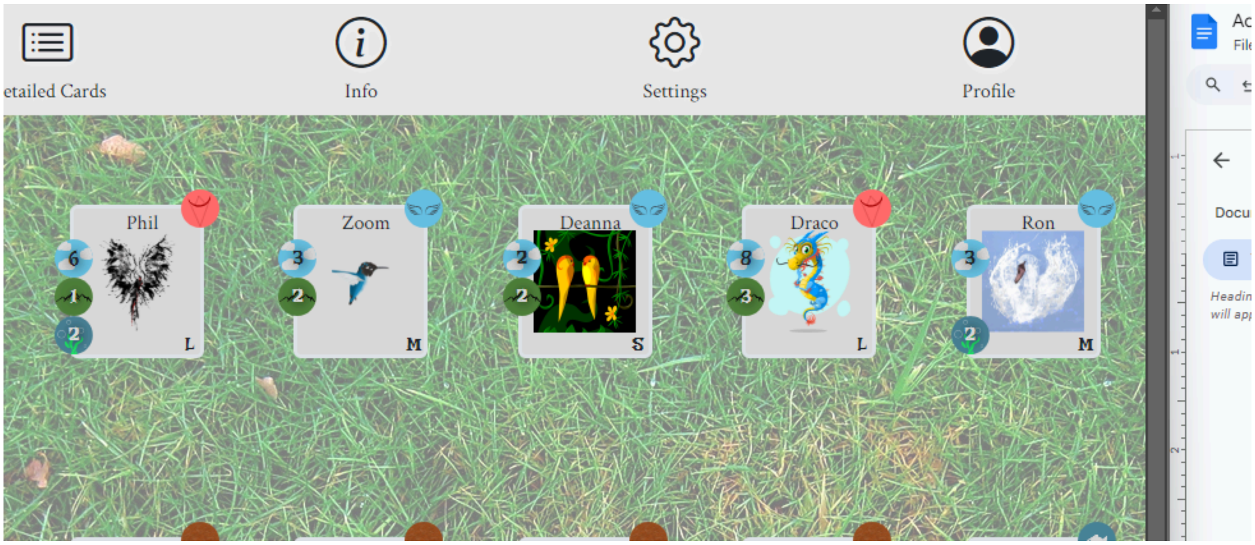
Charybdis Whirlpool

1

8

Eat & Recruiting - do the same to the top of the deck

Benson Bird large



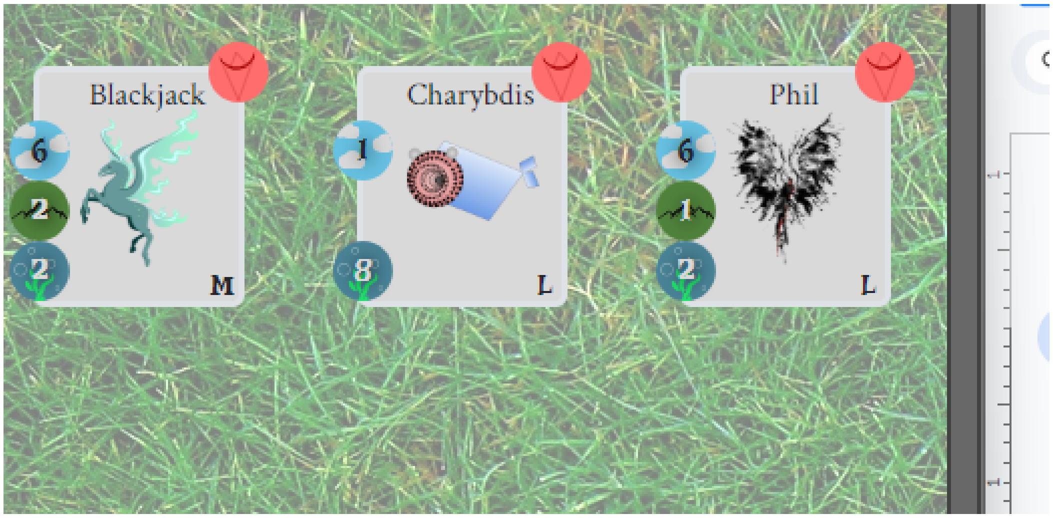
Phil 6 1 2 L

Zoom 3 2 M

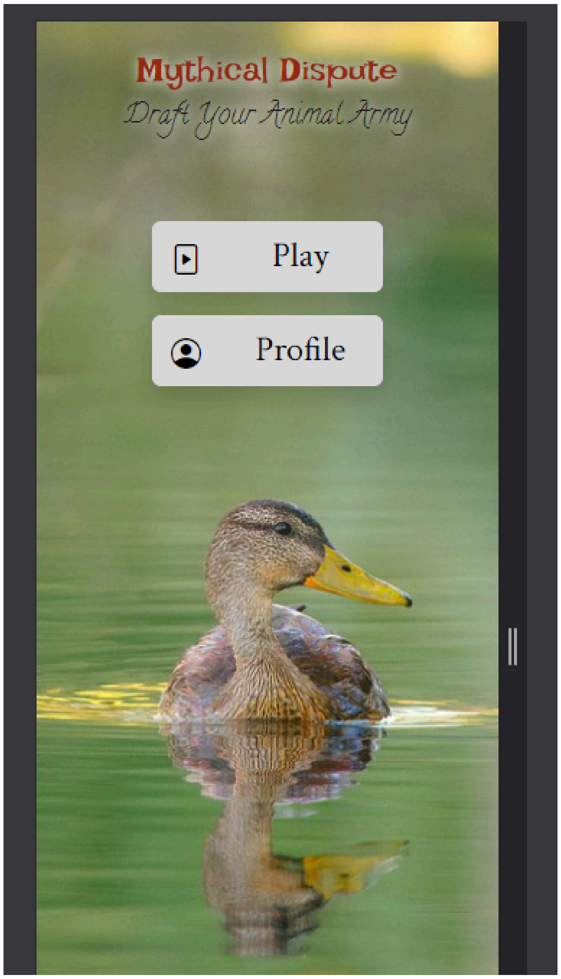
Deanna 2 2 S

Draco 8 3 L

Ron 3 2 M



Professional, organized and smooth experience


Mythical Dispute
Draft Your Animal Army
Play
Profile