

[Jump to bottom](#)

## Problems-Short: Initial Commit without DB logic #3

 **mgreiler** wants to merge 1 commit into `main` from `redeem-app-pr-1-problems-short` 

Conversation	0	Commits	1	Checks	0	Files changed	5
--------------	---	---------	---	--------	---	---------------	---



Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾ ⚙ ▾

2 .gitignore

```
...      @@ -0,0 +1,2 @@
1      + .DS_Store
2      + node_modules
```

694  redeem-app/package-lock.json 

Large diffs are not rendered by default.

15  redeem-app/package.json 

```
... @@ -0,0 +1,15 @@
1 + {
2 +   "name": "redeem",
3 +   "version": "1.0.0",
4 +   "description": "",
5 +   "main": "server.js",
6 +   "scripts": {
7 +     "start": "node server.js",
8 +     "test": "echo \"Error: no test specified\" && exit 1"
9 +   },
10 +   "author": "",
11 +   "license": "ISC",
12 +   "dependencies": {
13 +     "express": "^4.17.1"
14 +   }
15 + }
```

49 ■■■■■ redeem-app/public/index.html

```
...      @@ -0,0 +1,49 @@
1      + <html>
2      +     <head>
3      +         <title>Redeem Credits</title>
4      +     </head>
```

```
5 + <body>
6 +   <form>
7 +     <div>
8 +       <input id="Username" value="Your username">
9 +       <br/>
10 +      <label for="credits">Redeem Credits:</label>
11 +      <input id="credits" value="0">
12 +    </div>
13 +    <div>
14 +      <button id="sendButton">Send</button>
15 +    </div>
16 +  </form>
17 +  <div id="statusText"></div>
18 +</body>
19 +<script>
20 +  const urlString = window.location.href;
21 +  const url = new URL(urlString);
22 +  const userId = url.searchParams.get("userId") || 0;
23 +
24 +  const sendButton = document.getElementById('sendButton');
25 +  sendButton.addEventListener('click', (e) => {
26 +    e.preventDefault();
27 +    const credits = document.getElementById("credits").value;
28 +    const username = document.getElementById("Username").value;
29 +    fetch('/redeem', {
30 +      method: 'POST',
31 +      headers: {
32 +        'Content-Type': 'application/json',
33 +      },
34 +      body: JSON.stringify({
35 +        userId: userId,
36 +        credits: credits,
37 +        username: username
38 +      })
39 +    })
40 +    .then(response => response.json())
41 +    .then(data => {
42 +      document.getElementById('statusText').innerHTML = data.status;
43 +    })
44 +    .catch((error) => {
45 +      document.getElementById('statusText').innerHTML = error.status;
46 +    });
47 +  });
48 +</script>
49 +</html>
```



118 redeem-app/server.js

```
...  ...  @@ -0,0 +1,118 @@
1 + const express = require("express");
2 + const app = express();
3 + app.use(express.json());
4 + const port = 3000;
```



mgreiler Pending

Owner

Author

Make the port configurable.



Reply...

```
5 +
6 + app.use(express.static(__dirname + "/public"));
```

```
7 +
8 + app.get("/", function (req, res) {
9 +   res.sendFile(__dirname + "/public/index.html");
10 + });
11 +
12 + app.post("/redeem", function (req, res) {
13 +   const credits = req.body.credits;
14 +   const userId = req.body.userId;
15 +   const username = req.body.username;
16 +   console.log(`redeeming ${credits} credits for user: ${username} (${userId})`);
17 +
18 +   try {
19 +     const hours = redeemCredits(credits, userId);
20 +     res.send({
21 +       status: `User ${username} redeemed ${credits} credits to get ${hours} hours.`,
22 +     });
23 +   } catch (err) {
24 +     res.status(400).send({
25 +       status: err,
26 +     });
27 +   }
28 + });
29 +
30 + app.listen(port, () => {
31 +   console.log(`API listening at http://localhost:${port}`);
32 + });
33 +
34 + /**
35 +  * The player can purchase gaming hours by redeeming
36 +  * credits. How many hours one credit is
37 +  * worth depends on the level of the player.
38 +  */
39 + function redeemCredits(credits, playerId) {
40 +   playerLevel = getPlayerLevel(playerId);
```

**mgreiler** Pending

Owner

Author

Declare variables with let or const.



Reply...

```
41 +   hours = convertCreditsToHours(playerLevel, credits);
42 +
43 +   redeemHoursToPlayerProfile(hours, credits, playerId);
44 +   return hours;
45 + }
46 +
47 + /**
48 +  * Returns the level of the player.
49 +  */
50 + function getPlayerLevel(playerId) {
51 +   levelQuery = "SELECT playerLevel FROM players WHERE playerId = " + playerId;
```

**mgreiler** Pending

Owner

Author

Use parameterized queries to prevent SQL injection.



Reply...

```
52 +   playerLevel = executeQuery(levelQuery);
53 +
54 +   return playerLevel;
```

```
55 + }
56 +
57 + /**
58 +  * Adds the purchased hours to the players game hours.
59 +  */
60 + function redeemHoursToPlayerProfile(hours, credits, playerId) {
61 +   hourQuery = "SELECT hours FROM players WHERE playerId = " + playerId;
62 +   oldHours = executeQuery(hourQuery);
63 +
64 +   updateQuery = "Update players SET hours = " + (oldHours + hours) + " WHERE playerId = " +
        playerId;
65 +
66 +   try {
67 +     executeQuery(updateQuery);
68 +   } catch (err) {
69 +     throw new Error("Could not add hours.");
70 +   }
71 +
72 +   chargeCreditsFromPlayer(credits, playerId);
73 + }
74 +
75 + /**
76 +  * Charges the player with the credits redeemed.
77 +  */
78 + function chargeCreditsFromPlayer(credits, playerId) {
79 +   creditQuery = "SELECT credits FROM players WHERE playerId = " + playerId;
80 +   oldCredits = executeQuery(hourQuery);
```

mgreiler

Pending

Owner

Author

Wrap all DB interactions in try-catch.



Reply...

```
81 +
82 +   updateQuery = "Update players SET credits = " + (oldCredits - credits) + " WHERE playerId = " +
        playerId;
83 +
84 +   try {
85 +     executeQuery(updateQuery);
86 +   } catch (err) {
87 +     throw new Error("Could not charge credits: " + err);
88 +   }
89 + }
90 +
91 + /**
92 +  * This method converts the credits to game hours. Gamers that
93 +  * have a lower gaming level get more game hours for their credits.
94 +  * For more advanced gamers, buying new credits is more expensive.
95 +  *
96 +  * Players of level less than 3, get 3 times the hours of their credit
97 +  * Players of level less than 8, get 1.5 times the hours of their credit
98 +  * Players of level higher than 8, get just the hours of the credit.
99 +  */
100 + function convertCreditsToHours(playerLevel, credits) {
```

mgreiler

Pending

Owner

Author

Fix logic for players at level 3.



Reply...

```
101 +   if (playerLevel < 3) {
102 +       return 3 * credits;
103 +   } else if (playerLevel > 3 && playerLevel <= 8) {
104 +       return 1.5 * credits;
105 +   } else {
106 +       return 1 * credits;
107 +   }
108 + }
109 +
110 + /** TODO: implement */
111 + function getRandomInt(max) {
112 +     return Math.floor(Math.random() * Math.floor(max));
113 + }
114 +
115 + /** TODO: implement */
116 + function executeQuery() {
117 +     return getRandomInt(13);
118 + }
```