Edit Code

Jump to bottom

Initial Commit without DB logic #1

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

Conversation 0 Commits 1 Checks 0 Files changed 5

Changes from **all commits** File filter Conversations Jump to

2 .gitignore

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

694 redeem-app/package-lock.json

Load diff

Large diffs are not rendered by default.

15 redeem-app/package.json

```
@@ -0,0 +1,15 @@
+ {
+   "name": "redeem",
+   "version": "1.0.0",
+   "description": "",
+   "main": "server.js",
+   "scripts": {
+     "start": "node server.js",
+     "test": "echo \"Error: no test specified\" && exit 1"
+   },
+   "author": "",
+   "license": "ISC",
+   "dependencies": {
+     "express": "^4.17.1"
+   }
+ }
```

49 redeem-app/public/index.html

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

```
+     <body>
+         <form>
+             <div>
+                 <input id="Username" value="Your username">
+                 <br/>
+                 <label for="credits">Redeem Credits:</label>
+                 <input id="credits" value="0">
+             </div>
+             <div>
+                 <button id="sendButton">Send</button>
+             </div>
+         </form>
+         <div id="statusText"></div>
+     </body>
+     <script>
+         const urlString = window.location.href;
+         const url = new URL(urlString);
+         const userId = url.searchParams.get("userId") || 0;
+ 
+         const sendButton = document.getElementById('sendButton');
+         sendButton.addEventListener('click', (e) => {
+                 e.preventDefault();
+                 const credits = document.getElementById("credits").value;
+                 const username = document.getElementById("Username").value;
+                 fetch('/redeem', {
+                     method: 'POST',
+                     headers: {
+                         'Content-Type': 'application/json',
+                     },
+                     body: JSON.stringify({
+                         userId: userId,
+                         credits: credits,
+                         username: username
+                     })
+                 })
+                 .then(response => response.json())
+                 .then(data => {
+                     document.getElementById('statusText').innerHTML = data.status;
+                 })
+                 .catch((error) => {
+                     document.getElementById('statusText').innerHTML = error.status;
+                 });
+         });
+     </script>
+ </html>
```

118 redeem-app/server.js

```
@@ -0,0 +1,118 @@
+ const express = require("express");
+ const app = express();
+ app.use(express.json());
+ const port = 3000;
+ 
+ app.use(express.static(__dirname + "/public"));
+ 
+ app.get("/", function (req, res) {
+   res.sendFile(__dirname + "/public/index.html");
+ });
+ 
+ app.post("/redeem", function (req, res) {
+   const credits = req.body.credits;
+   const userId = req.body.userId;
```

```
+   const username = req.body.username;
+   console.log(`redeeming ${credits} credits for user: ${username} (${userId})`);
+
+   try {
+     const hours = redeemCredits(credits, userId);
+     res.send({
+       status: `User ${username} redeemed ${credits} credits to get ${hours} hours.`,
+     });
+   } catch (err) {
+     res.status(400).send({
+       status: err,
+     });
+   }
+ });
+
+ app.listen(port, () => {
+   console.log(`API listening at http://localhost:${port}`);
+ });
+
+ /**
+  * The player can purchase gaming hours by redeeming
+  * credits. How many hours one credit is
+  * worth depends on the level of the player.
+  */
+ function redeemCredits(credits, playerId) {
+   playerLevel = getPlayerLevel(playerId);
+   hours = convertCreditsToHours(playerLevel, credits);
+
+   redeemHoursToPlayerProfile(hours, credits, playerId);
+   return hours;
+ }
+
+ /**
+  * Returns the level of the player.
+  */
+ function getPlayerLevel(playerId) {
+   levelQuery = "SELECT playerLevel FROM players WHERE playerId = " + playerId;
+   playerLevel = executeQuery(levelQuery);
+
+   return playerLevel;
+ }
+
+ /**
+  * Adds the purchased hours to the players game hours.
+  */
+ function redeemHoursToPlayerProfile(hours, credits, playerId) {
+   hourQuery = "SELECT hours FROM players WHERE playerId = " + playerId;
+   oldHours = executeQuery(hourQuery);
+
+   updateQuery = "Update players SET hours = " + (oldHours + hours) + " WHERE playerId = " +
  playerId;
+
+   try {
+     executeQuery(updateQuery);
+   } catch (err) {
+     throw new Error("Could not add hours.");
+   }
+
+   chargeCreditsFromPlayer(credits, playerId);
+ }
+
+ /**
+  * Charges the player with the credits redeemed.
+  */
  function chargeCreditsFromPlayer(credits, playerId) {
```

```
79  +   creditQuery = "SELECT credits FROM players WHERE playerId = " + playerId;
80  +   oldCredits = executeQuery(hourQuery);
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) {
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 + }
```