**TITLE:**

BRAIN HUNT

**SOFT SKILLS:** Problem Solving, Thinking, Eye for details, Curiosity.

**HOW IT WORKS:**

**FOR USERS:**

BRAIN HUNT is an interesting puzzle web based game for kids. It has 5 levels.

To start the game, you should be logged in. After logging in. When you click start game button on the main header, you are redirected to the 1st Level of the game. For each level you have 2 tries, if you get 2 wrong answers the game ends. And as you give the correct answer within 2 tries you move to next level. As the level increases the difficulty level increases.

When the game ends, your total score of the current game is displayed and to update your overall score you need to click on update score button which will update your total score, accuracy and highest-level reached. Also, by clicking ‘My Stat’ in the header displays your updated scorecard.

**METHOD OF EVALUATION:**

It is on the basis of number of levels cleared.

Score: Each level cleared gives 4 points.

Accuracy: Percentage of points earned.

Highest Level Reached: Out of all games played it shows the maximum level reached.

Problem Solving: It is assessed from Level 1.

Thinking: It is assessed from Level 2 and Level 4.

Eye for details: It is assessed from Level 3 and Level 5.

Curiosity: It is assessed from the number of games played.

**FOR ADMIN:**

For admin there is only one Admin-Id which will be predefined. After clicking dashboard button the main header when admin is logged in, he admin is redirected to the dashboard. It displays Score, Accuracy and highest level of all the users along with the username.

If admin wants to change the password and the Admin-Id, it can be done:

In Backend -> controllers -> admin-controllers.js

On line 12 the id and passwords can be changed.

**TECHNOLOGIES USED:** MERN

**SETUP:**

Firstly, download fronted and backend.

Run **npm install** in both folders to install node modules.

Install the following dependencies:

**Frontend:**

nodemon, react-router-dom, react-dom, react-transition-group, react-scripts

Change env file by adding your server link where ‘your\_url’ is written.

**Backend:**

body-parser, mongoose, express-validator, multer, uuid, express, mongoose-unique-validator, jsonwebtoken, bcryptjs, axios

Change nodemon.json file by changing ‘your\_username’ to your database userId, ‘your\_password’ to your database password and ‘your\_databasename’ to the name of your project in the database.

In backend change mongodb driver connect link in app.js file to connect your database.

To compile and start run **npm start** in both frontend and backend.

**FEATURES:**

* This game has a secured login and sign. Only authenticated users can login into the game. And all users with email id can sign up into it. All the data stored is secured, password stored into the database is encrypted.
* This game has 5 levels which has multiple options to select. Each level has only 1 correct answer, which when selected takes the user to the next level. And if the user selects 2 incorrect choices in a single level gives the dead end to the user i.e., the game ends at that level.
* For every user data is maintained into the database which includes {userId, email, password, image(profile img), totalgamesplayed, score, accuracy, highestlevel). After each game played the score and accuracy and highest level are updated accordingly.
* For admin, the dashboard component displays the progress of all the users which includes: Username, Profile Image, Games Played, Score, Accuracy, Highest level Reached.
* An additional feature ‘My Stat’ is added which displays the scorecard (Games Played, Score, Accuracy, Highest Level Reached) of the logged in user along with its profile image.

**•**  For mobile users, a Side Drawer is maintained which contains the elements of the Main Header of the application (login, start game buttons etc.).