

Guess the password!

Provide a client/server solution for a game.

Client - HTML App

On page load, it will display an 8 digit string, you have to guess the right password to win.

Example:

- hint: 12345678, real answer: 87654321

Guess The Password!

Hint →

1 2 3 4 5 6 7 8

User Attempt 1 →

6 7 8 3 4 5 1 2

User Attempt 2 →

5 7 6 3 4 1 2 8

•

•

•

Type here

SUBMIT

Guide - HTML App

Highlight individual characters that are guessed correctly.

Mobile Friendly

Server - REST API

GET - /new-password

Response - { json }

Description - Return a random string of 8 unique digits

Sample Response:

```
{  
  hint: '12345678'  
}
```

Guide

Password has 8 unique digits. But server will shuffle the digits and return it as 'hint'

Only contains digits. E.g. [0-9]

Hint must be different from the actual password.

POST - /verify-password
Response - { json}
Description - Return details for a password guess
Sample Request Body: {json}

```
{  
  hint: '12345678',  
  answer: '67834512'  
}
```

Sample Response for a **right** answer:

```
{  
  correct: true,  
  hint: '12345678',  
  answer: '87654321'  
}
```

Sample Response for a **wrong** answer:

```
{  
  correct: false,  
  highlight: ['7', '4']  
  hint: '12345678',  
  answer: '67834512'  
}
```

Guide

'hint' should persist in server. Ideally using memory storage. Return 404 if a hint has no corresponding password stored in server.

Response tells UI which characters are correct so UI can highlight them.

General Guide:

- Feel free to change and design the UI however you want it.
- Use Node.js for server. React.js for front end. Use NPM.
- Feel free to design data model properties the way you want it.
- Some useful unit tests.
- No need for perfect error handling. Focus on code readability and structure.
- You can zip the source code or upload to github etc.