NullPointer documentation

NullPointer is a post-based website / forum in which users can post questions, rate, and comment the posts/comments of other people. It is built around MERN-stack and utilizes a front-end framework called Booststrap.

NullPointer features:

- Minimalistic and gorgeous data-driven design.
- Responsive layout with support for mobile devices.
- Fast user interface made with React.
- Authentication.
- Personal accounts and user pages with custom images.
- The ability to post topics and comment them.
- The ability to edit and delete posts and comments.
- Code highlighting in posts.
- Timestamps.
- Admin account.
- Full keyboard-input only support.
- Support for most modern browsers.

On backend NullPointer uses Node.js runtime with express.js backend framework. It uses the port 3000 as a default port if no port is defined in the environmental variables.

Regarding security, Nullpointer uses passport.js and json web tokens as a mean of authentication on the backend. Passwords are hashed using bcryptjs before stored in the database. User inputs are validated using express-js and dangerous symbols are regexed out or escaped (to prevent xss attacks).

Nullpointer stores all its data to mongodb via mongoose. Nullpointer uses Multer to handle image saving and encoding.

On frontend NullPointer utilizes React.js framework with React-Bootstrap. This allows the speed and scalability of NullPointer’s UI.

For date management NullPointer uses Luxon for its simplicity and lightness.

NullPoint also uses React-Helmet to handle all the title changes throughout the website.

For highlighting, NullPointer uses react-highlight, a wrapper package for highlight.js.

Installation and usage instructions:

1. Clone the repository (or download zip).
2. Open terminal in the root folder.
3. Type in “npm i” or “npm install”.
4. After installation completes, type in “npm run build”.
5. After building the client, type in “npm start”.

The server automatically generates an admin account if one doesn’t exist.

Admin credentials:

- Name: “admin”
- Password: “admin”

I’m aiming for the max points (50) from this project. I believe that my project is qualified to achieve that even though some elements would need some refining and code could be cleaned up a bit. Furthermore, some UI elements could have some polishing. I calculated the points using the following table.

Points:

Feature	Points
Predefined features:	
Basic features with well written documentation	25
Users can edit their own comments/posts	4
Utilization of a frontside framework, such as React, but you can also use Angular, Vue or some other	5
Use some highlight library for the code snippets, for example https://highlightjs.org/	2
Admin account with rights to edit all the post and comments and delete content (if a post is removed, all its comments should be removed too)	3
Test software for accessibility; can it be used only with keyboard / voice command? Can screen readers work with your application?	3
Provide a search that can filter out only those messages that have the searched keyword	2
Vote (up or down) posts and comments (only one vote per user)	3
User profiles can have images which are show next to posts/comments	3
User can click username and see user profile page where name, register date, (user picture) and user bio is listed	2
Last edited timestamp is stored and shown with posts/comments	2

Own features:	
Users are able to edit their own profile pages	2
Total:	56 (54 without own features)

Tests:

Test description	**Expected result**	**Test results**
Website accessibility using only a keyboard	The website should be perfectly functional without a mouse. Every feature should be accessible and work as normal.	The website worked perfectly using only a keyboard. The navigation was done using Tab and shift + tab and inputs using enter. Every feature was accessible and worked as expected.
Website accessibility using voice controls (tested on windows 10 voice controls)	The website should function as normal using voice commands as an input device.	The website was not usable. Only the menu navigation buttons were "clickable" but none of the poster or user previews were. However, a user is able to edit his profile using voice commands.
Screen reader functionality on the website (tested on windows 10 narrator).	The website should be narrated adequately to be used without vision.	Some parts of the website are usable without vision, however, some elements such as the password-hiding buttons are not narrated descriptively and may cause confusion without vision.