Design Decisions
This repository is an implementation of a basic user authentication system using bcrypt.js and JSON Web Tokens (JWT) with Node.js and Express framework. The server.js creates an HTTP server listening on port 3000 and has two endpoints for user registration and user login. Once a user registers, their hashed password is stored in memory. When a user logs in, the code verifies if the username exists and if the hashed password matches the stored password. If successful, it generates a JWT and returns it to the client. The code also has an endpoint for displaying the list of registered users, which can only be accessed with a valid JWT.

Thought Process
The primary goal was to develop a secure user authentication system that would protect sensitive user information with token. Bcrypt.js was used to securely store passwords, and JSON Web Tokens were used for secure token-based authentication.
The code has two endpoints, one for registration and another for login. Once registered, the user's information is stored in memory, and only the hashed password is stored. When logging in, the code retrieves the stored user information and compares the hashed password with the inputted password. If the password is correct, a JWT is generated, and the user can access the protected endpoint, which is /userlist.

Challenges
One challenge encountered was handling errors. The server will catch errors thrown by bcrypt and jwt libraries and returns appropriate error messages to the client.

Trade-offs:
A trade-off was made to use in-memory storage to store user information. In-memory storage may not be scalable or persistent for large-scale applications.

Summary
I implemented APIs providing a basic and secure user authentication system with Node.js. The use of bcrypt and JWT helps me to protect user_list and ensure secure communication between the client and server. Some additional features such as scalable data storage can be added for more complex applications in the future.