SAARTHI.AI Assessment

By Nalin Gupta

**Applying for Backend Engineer (INTERN)**

**Project is hosted on Heroku at** : <https://backend-assign.herokuapp.com/>

# **Table of Contents**

|  |  |
| --- | --- |
| **Serial Number** | **Title** |
| 1 | Project Description |
| 2 | Features |
| 3 | Architecture |
| 4 | Technologies Uses |

# **Project Description**

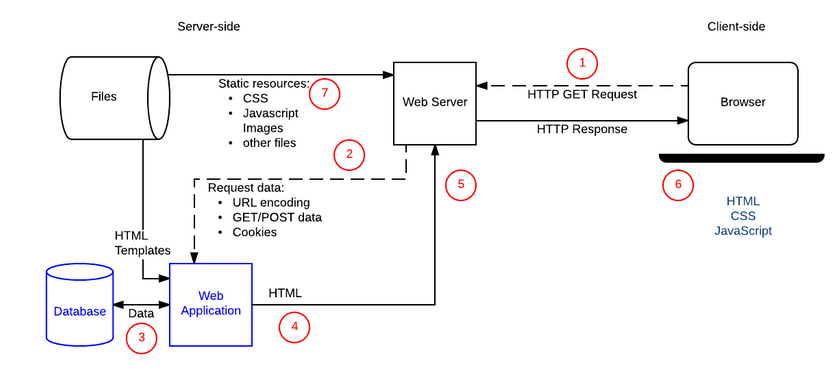
An application that scrapes text Content(paragraphs) from other websites and displays it. This app uses Node.js /Express /MongoDB with passport-local for authentication and puppeteer for scraping data.

# **Features**

This project consists of three screens:

1. Screen 1 (Login Page): This is a user login page where either a new user can register with the help of register link or an existing user can login via username and password. Various validations are applied on username and password fields.
2. Screen 2 (Register Page): This page consists of three input fields namely Username , Password and Confirm Password. It displays any error messages that may occur due to incomplete information or password mismatch while the user is entering their details. This page also consists a link that takes already registered users back to the login screen.
3. Screen 3 (Main Page) : This is the main page of the application that a user will land on upon successful authorization. Following features are available on this page:
   1. An input field that takes in the URL of the website whose text content the user wishes to see.
   2. A button that enables the user to submit their URL.
   3. A Logout button that logs out the user from the application and takes them back to the login page.
   4. A text box that renders on the page upon successful scraping of text content from the requested URL.
   5. Back-To-Top button that takes the user to the top of the page.

# **Architecture :**



*Image source : https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\_steps/Client-Server\_overview*

Requests are forwarded (2) to server-side code (as shown in the diagram). The server interprets the request, reads required information from the database (3), combines the retrieved data with HTML templates (4), and sends back response containing the generated HTML (5, 6)

# **Technologies Used :**

The following Tools & Technologies were used for achieving the project objectives-

Front End :

1. HTML
2. CSS
3. JavaScript
4. Materialize

Back End :

1. Node.js
2. Express Framework
3. MongoDB
4. Ejs as View Engine

Dependencies Uses :

1. Express , Express session
2. mongoose
3. Passport , passport-local
4. Puppeteer

Dev Dependencies - Nodemon