

Mini Project Report

on

BUILDLINK Construction Procurement Platform

Submitted by

CHIRIYANKANDATH MELBIN SEBASTIAN (20221095)

EFFIN JOE WILSON (20221096)

SIDHU SANGEERTH S (20321082)

TOVIN THANKACHAN THOMAS (20321090)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology
in Computer Science and Engineering.



DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

JULY 2024

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a Mini Project Report titled

BUILDLINK
Construction Procurement Platform

Submitted by

CHIRIYANKANDATH MELBIN SEBASTIAN (20221095)
EFFIN JOE WILSON (20221096)
SIDHU SANGEERTH S (20321082)
TOVIN THANKACHAN THOMAS (20321090)

of VI Semester, Computer Science and Engineering in the year 2024 in partial fulfillment
requirements for the award of degree of Bachelor of Technology in Computer Science and
Engineering of Cochin University of Science and Technology.

Dr. Sudheep Elayidom M
Head of Division

Mrs.Preetha S
Project Coordinator

Mrs.Sheena S
Project Guide

Acknowledgement

We would like to take this opportunity to express our heartfelt gratitude to the supreme being, the source of all knowledge, whose blessings have been our guiding light throughout this endeavour. It is with immense pleasure that we acknowledge the contributions of the individuals who have made this mini project Buildlink a success. We extend our deepest appreciation to Prof. Mrs. Sheena S, our project guide, whose unwavering guidance and constant support have been instrumental in shaping this project. Her invaluable inputs and suggestions have played a crucial role in the implementation and functionality of the Buildlink. This project would not have reached its full potential without her expertise and dedication. Furthermore, we extend our heartfelt thanks to Prof. Dr. Sudheep Elayidom M, Head of the Computer Science and Engineering division, for her unwavering encouragement and support.

Declaration

We, Chiriyankandath Melbin Sebastian, Effin Joe Wilson, Sidhu Sangeerth S, and Tovin Thankachan Thomas hereby declare that this mini project is the record of authentic work carried out by us during the academic year 2021-2025 and has not been submitted to any other University or Institute towards the award of any degree.

CHIRIYANKANDATH MELBIN SEBASTIAN (20221095)

EFFIN JOE WILSON (20221096)

SIDHU SANGEERTH S (20321082)

TOVIN THANKACHAN THOMAS (20321090)

Abstract

BuildLink is a comprehensive web-based platform engineered to revolutionize the construction industry by enabling seamless collaboration and resource exchange among stakeholders in the procurement process. It addresses the needs of two primary user groups: Users—comprising contractors, project managers, and other construction professionals—who seek essential resources such as raw materials, heavy machinery, and skilled labor; and Agents—including suppliers, equipment providers, and service professionals—who offer these critical resources.

At the heart of BuildLink is its commitment to efficiency, accessibility, and transparency. The platform’s user-friendly interface is designed to simplify the procurement process, making it easier for Users to find exactly what they need through advanced search functionalities. Users can quickly locate materials, equipment, or services, while Agents can efficiently showcase their offerings to a targeted audience, ensuring that resources are matched with demand in real-time.

By centralizing the procurement process, BuildLink enhances productivity across all stages of construction. It minimizes resource wastage by ensuring that materials and services are utilized where and when they are needed most. Additionally, the platform’s data-driven insights allow for better decision-making, enabling Users to optimize their resource allocation and reduce costs.

BuildLink’s centralized nature also fosters greater transparency within the construction supply chain. All transactions, communications, and resource exchanges are logged and accessible, providing all parties with a clear overview of their operations. This transparency builds trust among Users and Agents, leading to more reliable partnerships and long-term collaboration.

Contents

Acknowledgement	i
Declaration	ii
Abstract	iii
Table of Contents	v
List of Figures	vi
1 Introduction	1
2 Literature Review	3
3 System Analysis	5
3.1 Existing System	5
3.2 Proposed System	6
4 System Study	8
4.1 Software Requirements Specification	8
4.1.1 Purpose	8
4.1.2 Product Scope	8
4.1.3 Project overview	9
4.1.4 Functional Requirements	9
4.1.5 Non Functional Requirements	10
4.2 System Implementation	11
4.2.1 Technology Stacks	11
4.3 System Architecture	13
4.3.1 User Interface	13

4.3.2	Backend Logic	13
4.3.3	Database	13
5	System Design	14
5.1	Introduction	14
5.2	Data Flow Diagrams	14
5.3	Use case Diagrams	14
5.4	Modular Design	15
5.5	Input Output Design	16
6	System Implementation	17
6.1	Sample Code	17
6.1.1	User Authentication Routes	17
6.1.2	User Authentication Helpers	18
6.1.3	Agent Service Management Routes	19
6.1.4	Agent Service Helper Functions	20
6.2	Screenshots	21
6.3	Test Case	25
7	Result	27
8	Future Scope	28
9	Conclusion	30
10	References	31

List of Figures

5.1	DFD	15
5.2	Use case diagram for agent and user	16
6.1	User Authentication Routes	17
6.2	User Authentication Helpers	18
6.3	Agent Service Management Route	19
6.4	Agent Service Helper Functions	20
6.5	Sign Up Page	21
6.6	Login Page	21
6.7	Home page	22
6.8	user dashboard	22
6.9	Add services page - Agent	23
6.10	verify agents - admin	23
6.11	Agent Listing	24
6.12	Test Case 1	25
6.13	Test Case 2	26

Chapter 1

Introduction

In an era where the construction industry is constantly evolving, fostering a seamless and efficient procurement process is paramount to driving progress and innovation. Traditional procurement mechanisms often fall short in providing the necessary transparency and ease of access required to connect users with the essential resources they need. However, a pioneering solution has emerged to address these concerns and empower the construction community: BuildLink.

BuildLink is a revolutionary web-based platform designed to transform the construction industry through cutting-edge technology and user-centric design. This platform caters to two primary user roles: Users seeking construction resources such as raw materials, machinery, and skilled labor, and Agents ready to provide these essential services. By prioritizing efficiency, accessibility, and transparency, BuildLink aims to revolutionize the construction procurement landscape.

At the core of BuildLink's innovative system lies its intuitive interface, which ensures that users can easily navigate the platform and find exactly what they need. Advanced search capabilities enable users to swiftly locate resources, while robust transaction features streamline the procurement process, reducing time and effort. BuildLink's commitment to transparency means that users can trust the platform to provide accurate and reliable information, fostering a sense of confidence and collaboration among all parties involved.

By bridging the gap between resource seekers and providers, BuildLink dismantles the barriers that often hinder the procurement process. Challenges

such as inefficient communication, lack of access to resources, and opaque transactions are addressed head-on, creating an environment where users can thrive. The platform empowers users to take control of their procurement needs, facilitating smoother operations and ultimately contributing to the growth and success of the construction industry.

This collaboration between users and agents is the cornerstone of BuildLink's mission—a proactive approach to construction procurement that involves active participation from all members of the industry. Empowered by this new system, construction professionals become an invaluable force in driving progress, innovation, and sustainability, thereby strengthening the very fabric of the industry.

As a result of this ground-breaking technology, BuildLink lays the foundation for a more efficient and connected future in construction. It fosters an environment where users are encouraged to take an active role in optimizing their procurement processes, without hesitation or complexity. The unparalleled efficiency and transparency offered by the platform revolutionize the way resources are exchanged, paving the way for transformative advancements in the construction sector.

In conclusion, BuildLink emerges as a beacon of progress, bridging the gap between resource seekers and providers in the pursuit of a more efficient and transparent construction industry. With its seamless integration of advanced technology and user-focused design, it instills confidence in those who wish to play a role in driving industry innovation. This revolutionary solution stands as a testament to the potential of technology in creating positive change, emphasizing that together, united in purpose, we can build an industry that thrives in efficiency and collaboration.

Chapter 2

Literature Review

BuildLink is a pioneering web-based platform aimed at revolutionizing the construction industry by facilitating seamless collaboration and resource exchange. This literature review explores foundational works in the fields of construction management, supply chain integration, digital marketplaces, and blockchain technology. The chosen literature sources are crucial to understanding the fundamentals of these areas and their potential applications in developing an efficient and transparent construction procurement system like BuildLink.

1. Construction Supply Chain Management: Concepts and Case Studies - Greger Lundesjö

Greger Lundesjö's work on construction supply chain management delves into the complexities of managing resources, logistics, and information flows in construction projects. The book presents various case studies and concepts essential to optimizing supply chain operations in the construction industry.

Key Concepts:

- Integrated supply chain management
 - Resource planning and logistics
 - Case studies on successful construction projects
- Relevance to BuildLink: Lundesjö's work provides insights into the challenges and solutions related to supply chain management in construction. These concepts are fundamental to BuildLink's mission of creating a streamlined, efficient procurement process by integrating resources and logistics effectively.

2. The Smart Construction Site - John E. Schaufelberger and Len Holm

This book explores the application of smart technologies in construction, emphasizing the use of digital tools to enhance project efficiency and collaboration. It discusses how emerging technologies can transform traditional construction practices. Key Concepts:

- Digital tools and technologies in construction
- Enhancing project efficiency through smart systems
- Collaboration and communication improvements
Relevance to BuildLink: The concepts of digital transformation in construction are directly applicable to BuildLink's platform, which aims to enhance collaboration and resource management through an intuitive interface and advanced search capabilities.

3. Digital Marketplaces Unleashed - Claudia Linnhoff-Popien, Ralf Schneider, and Michael Zaddach

This book discusses the rise of digital marketplaces and their impact on various industries. It explores how online platforms can facilitate transactions, improve transparency, and create value for users. Key Concepts:

- Structure and function of digital marketplaces
- Enhancing transparency and trust in online transactions
- Case studies on successful digital platforms
Relevance to BuildLink: The principles outlined in this book are essential for understanding how BuildLink can create a transparent and efficient marketplace for construction resources. The emphasis on trust and transaction efficiency aligns with BuildLink's goals.

Chapter 3

System Analysis

3.1 Existing System

Before the introduction of BuildLink, traditional construction procurement systems were commonly used. These systems relied on manual methods for resource procurement, communication, and transaction management, often leading to inefficiencies and lack of transparency. Users faced challenges in finding and verifying resources, and transactions were often slow and prone to errors.

Functionalities of the Existing System:

1. **Manual Procurement:** Users procured construction resources through phone calls, emails, or physical visits to suppliers, which was time-consuming and often led to delays.
2. **Limited Transparency:** Traditional systems lacked transparency, making it difficult for users to verify the authenticity and availability of resources.
3. **Paper-Based Documentation:** Transactions and communications were often documented manually on paper, leading to errors, loss of information, and inefficiencies in data management.
4. **Inefficient Communication:** Communication between resource seekers and providers was often fragmented and inefficient, leading to misunderstandings and delays.
5. **Lack of Integrated Platform:** There was no integrated platform to

streamline the procurement process, making it difficult for users to manage their resources and transactions effectively.

The existing traditional construction procurement system lacked efficiency, transparency, and effective communication channels. BuildLink aims to overcome these limitations by providing a centralized, efficient, and transparent platform for construction resource procurement.

3.2 Proposed System

BuildLink is a web-based platform designed to revolutionize the construction industry by facilitating seamless collaboration and resource exchange between users and agents involved in the procurement process. The proposed system leverages advanced technology and user-centric design to create a transparent, efficient, and secure platform.

Functionalities of the Proposed System:

1. **Centralized Procurement Platform:** BuildLink provides a centralized platform for users to procure construction resources, streamlining the procurement process and reducing delays.
2. **Enhanced Transparency:** The platform ensures transparency by providing verified information about resources, suppliers, and transactions, enabling users to make informed decisions.
3. **Digital Documentation:** Transactions and communications are documented digitally, reducing errors and improving data management efficiency.
4. **Advanced Search Capabilities:** The platform features advanced search capabilities, allowing users to quickly find the resources they need based on various criteria.
5. **User-Friendly Interface:** The platform's intuitive interface makes it easy for users to navigate and manage their procurement processes effectively.

Key Components of the Proposed System:

1. **User Interface:** A user-friendly frontend that allows users to easily navigate the platform, search for resources, and manage transactions.

2. **Backend Dashboard:** An administrative dashboard that enables administrators to manage and monitor transactions, resources, and user interactions.
3. **Advanced Search Engine:** A powerful search engine that allows users to find resources quickly and efficiently based on specific criteria.

By leveraging advanced technology and user-centric design principles, BuildLink aims to create a robust, efficient, and transparent platform for construction resource procurement. This system not only enhances the efficiency of the procurement process but also fosters collaboration and trust among users, ultimately contributing to the growth and success of the construction industry.

Chapter 4

System Study

4.1 Software Requirements Specification

Purpose

BuildLink is a web-based platform aimed at revolutionizing the construction industry by facilitating seamless collaboration and resource exchange between users and agents involved in the procurement process. The platform caters to two primary user roles: Users seeking construction resources such as raw materials, machinery, and skilled labor, and Agents ready to provide these essential services. BuildLink prioritizes efficiency, accessibility, and transparency in the construction procurement landscape through its intuitive interface, advanced search capabilities, and robust transaction features.

Product Scope

Target Users:

- **Users:** Construction professionals, contractors, or individuals seeking resources for construction projects. This could include materials (lumber, concrete, etc.), machinery (excavators, cranes), and skilled labor

(electricians, plumbers, carpenters).

- **Agents:** Suppliers of construction materials and equipment, as well as skilled labor providers (construction companies, staffing agencies, independent contractors).

Core functionalities:

- **User & Agent Matching:** A user-friendly platform that connects users with relevant agents based on their specific needs. This likely involves searchable databases and advanced filtering options.
- **Resource Discovery:** Users can search for a wide range of construction resources, including materials, equipment, and skilled labor.
- **Verification Process:** BuildLink might have a system to verify the legitimacy and qualifications of agents (licensed suppliers, insured contractors, etc.) to ensure user confidence.
- **Transparency:** Focus on clear pricing information and communication to foster trust and fair dealings between users and agents.

Project overview

The project contains following functions:-

Functional Requirements

Authentication:

Users and Agents must be able to register and log in to their respective accounts.

Authentication mechanisms should ensure security and data privacy.

Security:

The platform should implement industry-standard security measures to protect user data and transactions.

Encryption should be used to secure sensitive information such as passwords and payment details.

Agent Dashboard:

Agents should have a dedicated dashboard to showcase their offerings, manage listings, and track transactions.

Resource Posting:

Agents should be able to post listings for construction resources including raw materials, machinery, and skilled labor. Listings should include detailed information about the resource, pricing, and availability.

Resource Search:

Users should be able to search for resources based on various criteria such as type, location, and availability.

The search functionality should be intuitive and provide relevant results.

Non Functional Requirements

Performance:

BuildLink should be able to handle a large volume of users and transactions without significant performance degradation.

Response times for search queries and transaction processing should be minimal.

Security:

The platform should implement industry-standard security measures to protect user data and transactions.

Encryption should be used to secure sensitive information such as passwords and payment details.

Usability:

The user interface should be intuitive and user-friendly, catering to users with varying levels of technical expertise.

Navigation should be straightforward, and key features should be easily accessible.

Reliability:

BuildLink should be highly reliable, with minimal downtime or system failures.

Regular backups of data should be performed to prevent data loss.

Constraints:

BuildLink will initially target users and agents within a specific geographic region.

The platform will rely on third-party services for payment processing and real-time tracking.

Future Enhancements:

Future developments may include expanding service offerings, enhancing real-time tracking capabilities, and integrating with additional third-party services.

Continuous improvement will be based on user feedback and emerging industry trends.

Glossary:

Users: Individuals or organizations seeking construction resources.

Individuals or organizations offering construction resources and services.

A user interface that provides access to various features and functionalities within the platform

4.2 System Implementation

Technology Stacks

Node.js

The backend of BuildLink is powered by Node.js, a robust JavaScript runtime environment that provides the necessary scalability and performance for handling a high volume of requests and real-time data exchange.

Express.js

Express.js is used as the web application framework for Node.js, offering a fast, unopinionated, and minimalist approach to building the server-side logic. It handles routing, middleware, and server-side rendering efficiently.

MongoDB

MongoDB serves as the primary database for BuildLink, providing a flexible and scalable solution for storing user and agent data, including resource details and transaction records. Its NoSQL structure allows for easy management of unstructured data.

Express-Handlebars

The front-end of BuildLink is rendered using Express-Handlebars, a powerful templating engine that facilitates dynamic content rendering. It integrates seamlessly with Express.js, allowing for efficient server-side rendering of web pages.

bcrypt

bcrypt is utilized for securely hashing and storing user passwords, ensuring robust security for user authentication and data protection.

express-fileupload

This middleware handles file uploads, allowing users and agents to upload necessary documents and images related to resources and services, which are then stored securely in the specified directories.

express-session

Session management in BuildLink is handled by express-session, enabling the platform to maintain user sessions and authentication states across different interactions.

Handlebars.js

Handlebars.js is employed for front-end templating, enabling the creation of dynamic and reusable components for the user interface, which enhances user experience and interaction.

hbs

The hbs package provides additional support for using Handlebars with Express, ensuring seamless integration and efficient rendering of templates.

MongoDB Native Driver

The MongoDB native driver is used to interact with the MongoDB database, providing a powerful and flexible API for querying and managing data.

morgan

Morgan is a logging middleware for Node.js that helps in monitoring HTTP requests and debugging issues, contributing to the overall stability and maintainability of the platform.

nodemon

Nodemon is used during development to automatically restart the server when file changes are detected, enhancing the development workflow and productivity.

objectid

The objectid package is used for generating and handling MongoDB ObjectIDs, ensuring unique identifiers for database records.

4.3 System Architecture

User Interface

The user interface of BuildLink is built using Handlebars templates rendered by Express.js, providing a responsive and user-friendly experience. The interface allows users to search for resources, view agent profiles, and initiate transactions seamlessly.

Backend Logic

The backend logic is implemented in Node.js using Express.js. It handles user authentication, resource management, transaction processing, and communication between users and agents. The logic also includes middleware for session management, file uploads, and error handling.

Database

MongoDB is used to store user data, agent profiles, resource details, and transaction records. The database schema is designed to be flexible, allowing for easy scaling and management of data.

Chapter 5

System Design

5.1 Introduction

Designing requires a careful planning and thinking on the part of the system designer. Designing a system means to plan how the various parts of it are going to achieve the desired goal. After the software requirements have been analysed and specified, design is the first of the three technical activities. Designing, coding and testing are required to build and verify the web application.

5.2 Data Flow Diagrams

Data Flow Diagram is a pictorial way of showing the flow of data in to/within the system, around the system and out of the system. It is a graphical representation of flow of data within a system. Unlike flowcharts, DFDs do not give detailed descriptions of modules but graphically describe data and how the data interacts with the system. The DFD enables us to visualize how the system operates, its final output and the implementation of the system as a whole including modification if any. The purpose of data flow diagram is to provide a semantic bridge between users and system developers.

5.3 Use case Diagrams

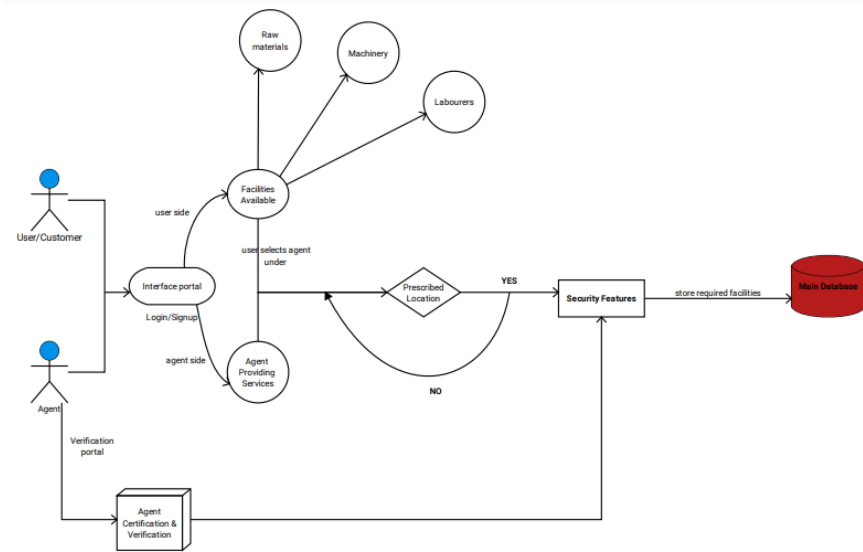


Figure 5.1: DFD

The Data Flow Diagram (DFD) outlines the workflow of a construction resource management system, focusing on interactions between Users and Agents. Users log in through an interface to access and select construction resources like raw materials and labor. Agents, who provide these resources, must first be certified and verified. Once a User selects an Agent, the system checks if the Agent is in the correct location. If verified, security checks are performed, and the transaction details are securely stored in the main database. The DFD emphasizes security, verification, and efficient resource management throughout the process.

A database is accessed, read from and written to, using an interface that allows applications to store and retrieve data. It includes a real-time NoSQL database for syncing and storing data across clients, authentication and user management features for easy user registration and login. It has an authentication, firestore database and a storage, which stores all the user datas.

5.4 Modular Design

User Side:

User Authentication module handles user registration, login, and authentication processes, Privacy and Security module ensures user privacy and implements security measures to protect user data.



Figure 5.2: Use case diagram for agent and user

The Use Case Diagram illustrates how Users and Agents interact with the BuildLink system. Users log in to search for resources, request quotes, place orders, and provide feedback. Agents log in to list resources, respond to quotes, confirm orders, and manage feedback. Additionally, Agents undergo a verification process via an admin panel. The system centralizes these interactions, focusing on specific locations to facilitate seamless resource transactions between Users and Agents.

5.5 Input Output Design

- Sign Up
Input : Username, mail id, password.
Output : The user is successfully registered if all the details provided are valid.
- Login
Input : mail id and password.
Output : Logged in successfully if the user is already present.
- Profile
Input : User can add about themselves and interests.
Output : All the changes will be visible on their profiles.
- Sign out
Input : Click on Sign out.
Output : Logged out successfully.

Chapter 6

System Implementation

6.1 Sample Code

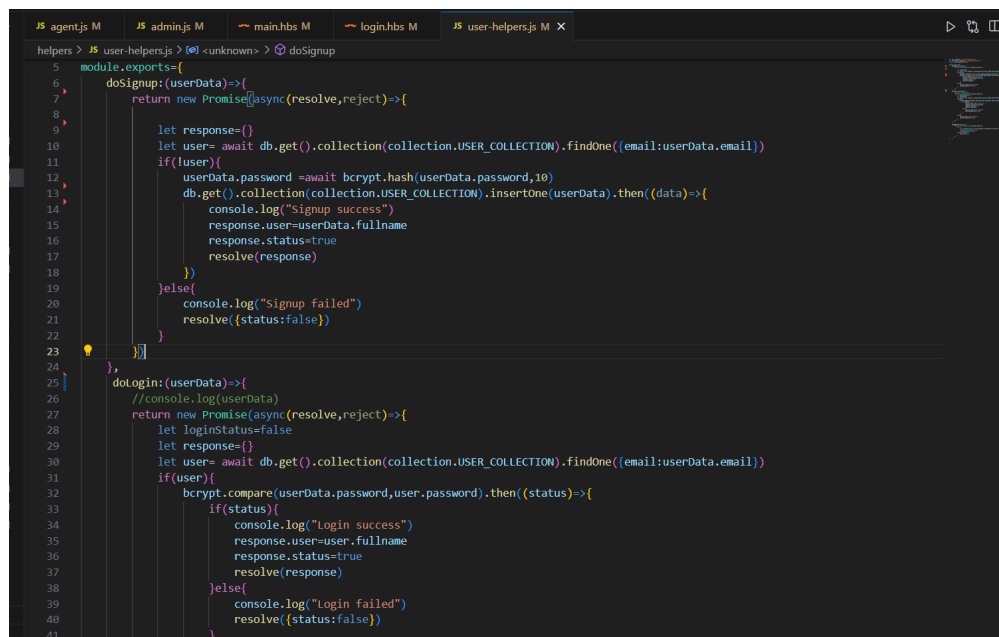
User Authentication Routes

```
routes > JS usersjs > router.post('/login') callback > then() callback
23 router.get('/login', function(req, res, next) {
24   //console.log("user")
25   if(req.session.user){
26     res.redirect('/user')
27   }else{
28     res.render('user/login',{loginErr:req.session.userloginErr})
29     req.session.userloginErr=false
30   }
31 });
32
33 router.post('/login', (req, res, next) =>{
34   //console.log("user loggedIn")
35   userHelpers.doLogin(req.body).then((response)=>{
36     if(response.status){
37       req.session.user=response.user
38       req.session.userLoggedIn=true
39       res.redirect('/user');
40     }else{
41       req.session.userloginErr='Invalid username or password'
42       console.log("Invalid username or password")
43       res.redirect('/user/login');
44     }
45   })
46 });
47
48 router.get('/signup', function(req, res, next) {
49   //console.log("user signup")
50   res.render('user/signup',{SignupErr:req.session.userSignupErr})
51   req.session.userSignupErr=false
52 });
53
54 router.post('/signup', (req, res)=> {
55   //console.log(req.body)
56   userHelpers.doSignup(req.body).then((response)=>{
57     if(response.status){
58       req.session.user=response.user
59       req.session.userLoggedIn=true
60     }
```

Figure 6.1: User Authentication Routes

The provided code defines routes for user login and signup. The /login GET route renders the login page if the user is not already logged in, otherwise, it redirects to the user page. The /login POST route handles login attempts, setting session data and redirecting based on the login success. The /signup GET route renders the signup page, while the /signup POST route handles user registration, checking for existing users and creating new ones, with appropriate session updates and redirects.

User Authentication Helpers



```
5 module.exports={
6   doSignup:(userData)=>{
7     return new Promise(async(resolve,reject)=>{
8
9       let response={}
10      let user= await db.get().collection(collection.USER_COLLECTION).findOne({email:userData.email})
11      if(!user){
12        userData.password =await bcrypt.hash(userData.password,10)
13        db.get().collection(collection.USER_COLLECTION).insertOne(userData).then((data)=>{
14          console.log("Signup success")
15          response.user=userData.fullname
16          response.status=true
17          resolve(response)
18        })
19      }else{
20        console.log("Signup failed")
21        reject({status:false})
22      }
23    })
24  },
25  doLogin:(userData)=>{
26    //console.log(userData)
27    return new Promise(async(resolve,reject)=>{
28      let loginStatus=false
29      let response={}
30      let user= await db.get().collection(collection.USER_COLLECTION).findOne({email:userData.email})
31      if(user){
32        bcrypt.compare(userData.password,user.password).then((status)=>{
33          if(status){
34            console.log("Login success")
35            response.user=user.fullname
36            response.status=true
37            resolve(response)
38          }else{
39            console.log("Login failed")
40            reject({status:false})
41          }
42        })
43      }
44    })
45  }
46 }
```

Figure 6.2: User Authentication Helpers

The doSignup function handles user registration by checking if the user already exists, hashing the password, and saving the user data to the database if the user does not exist. The doLogin function handles user login by verifying the email and password, returning the user data if authentication is successful, and updating the session status accordingly.

Agent Service Management Routes

```

JS agent.js x JS admin.js JS user.js M JS app.js M main.hbs M index.hbs M JS admin-helpers.js JS user-helpers.js M
routes > JS agent.js > router.post('/update-profile') callback > then() callback
70
71 router.get('/my-services',verifyLogin,(req,res)=>{
72   var name = req.session.agent.name
73   let agentId=req.session.agent.agentId
74   agentHelpers.doTakeAgentServices(agentId).then((agentServices)=>{
75     //console.log(agentServices)
76     res.render('agent/my-services',{agent:true,name,agentServices});
77   })
78 })
79
80 router.get('/add-services',verifyLogin,(req,res)=>{
81   var name = req.session.agent.name
82   res.render('agent/add-services',{agent:true,name});
83 })
84
85 router.post('/add-services',(req,res)=>{
86   //console.log(req.session)
87   let agentId=req.session.agent.agentId
88   agentHelpers.addServices(req.body,agentId).then(()=>{
89     res.redirect('/agent/my-services')
90   })
91 })
92
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
node + - - - - -
GET /agent/login 200 56.270 ms - 2814
GET /stylesheets/style.css 304 4.391 ms - -
GET /javascripts/script.js 304 5.430 ms - -
Login success
POST /agent/login 302 146.752 ms - 56
GET /agent 200 22.786 ms - 4706
GET /stylesheets/style.css 304 2.522 ms - -
GET /images/profile.jpeg 304 3.034 ms - -
GET /javascripts/script.js 304 1.268 ms - -
GET /agent/my-services 200 45.349 ms - 7386
GET /stylesheets/style.css 304 3.114 ms - -
GET /images/profile.jpeg 304 2.935 ms - -
GET /javascripts/script.js 304 2.088 ms - -

```

Figure 6.3: Agent Service Management Route

these routes are for managing agent services. The `/my-services` GET route fetches services associated with the logged-in agent using their `agentId` from the session and renders the `my-services` view, displaying the agent's services. The `/add-services` GET route renders the `add-services` view with the agent's name to add new services. The POST route `/add-services` adds new services submitted via a form to the database for the logged-in agent and redirects back to `/my-services`.

Agent Service Helper Functions

```

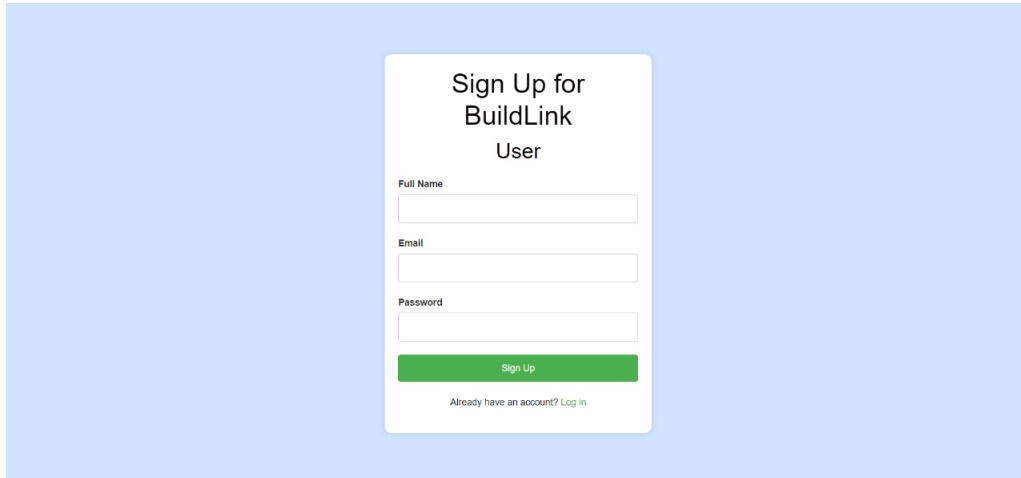
helpers > $ agent-helpers.js > doTakeAgentServices
6   module.exports = {
62   addServices: (services, agentId) => {
64       return new Promise(async(resolve, reject) => {
65           let agentService = await db.get().collection(collection.AGENT_SERVICES_COLLECTION).findOne({ agent: new ObjectId(agentId) })
66           if (!agentService) {
67               let agentObj = {
68                   agent: new ObjectId(agentId),
69                   services: Array.isArray(services) ? services : [services] // Ensure services is an array
70               };
71               db.get().collection(collection.AGENT_SERVICES_COLLECTION).insertOne(agentObj).then((response) => {
72                   resolve();
73               });
74           } else {
75               //console.log("Already Existing");
76               // Update the existing agent service with the new services
77               db.get().collection(collection.AGENT_SERVICES_COLLECTION).updateOne(
78                   { agent: new ObjectId(agentId) },
79                   { $push: { services: { $each: Array.isArray(services) ? services : [services] } } })
80               .then((response) => {
81                   resolve();
82               });
83           }
84       });
85   },
86
87   doTakeAgentServices: (agentId) => {
88       return new Promise(async(resolve, reject) => {
89           try {
90               let agentServices = await db.get().collection(collection.AGENT_SERVICES_COLLECTION).findOne({ agent: new ObjectId(agentId) })
91               //console.log(JSON.stringify(agentServices));
92               resolve(agentServices);
93           } catch (error) {
94               reject(error);
95           }
96       });
97   },
98 }

```

Figure 6.4: Agent Service Helper Functions

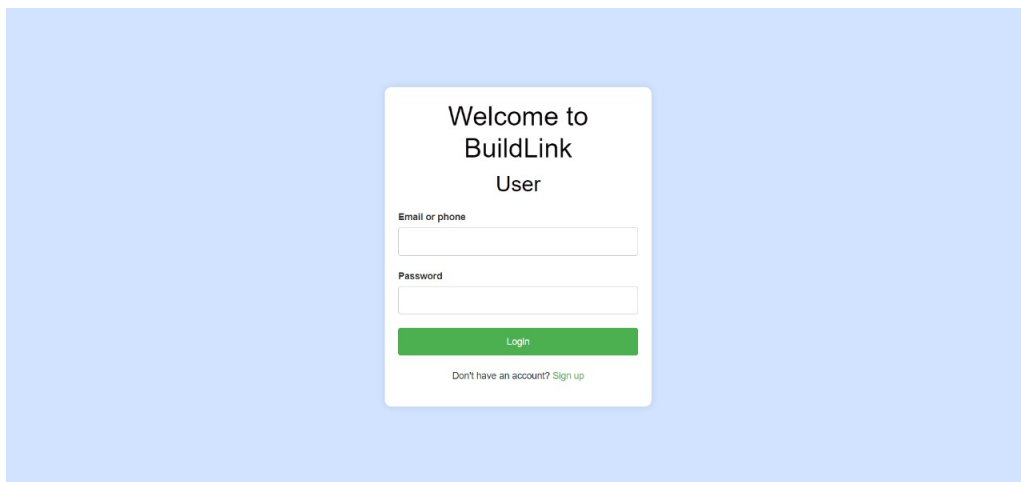
The `addServices` function adds services to the database for a given `agentId`. It checks if services for the agent already exist; if not, it creates a new entry. If services exist, it updates the existing entry by appending new services to the services array. The `doTakeAgentServices` function retrieves and returns the services associated with a given `agentId` from the database, facilitating display and management of agent services in the application.

6.2 Screenshots



A screenshot of a sign-up form titled "Sign Up for BuildLink User" centered on a light blue background. The form is a white card with rounded corners. It contains three input fields: "Full Name", "Email", and "Password". Below the "Password" field is a green "Sign Up" button. At the bottom of the card, there is a link that says "Already have an account? Log in".

Figure 6.5: Sign Up Page



A screenshot of a login form titled "Welcome to BuildLink User" centered on a light blue background. The form is a white card with rounded corners. It contains two input fields: "Email or phone" and "Password". Below the "Password" field is a green "Login" button. At the bottom of the card, there is a link that says "Don't have an account? Sign up".

Figure 6.6: Login Page

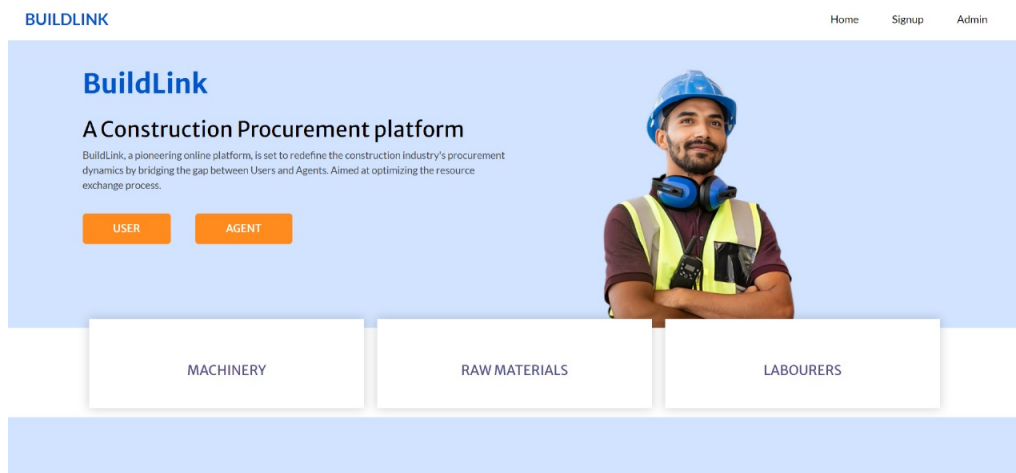


Figure 6.7: Home page

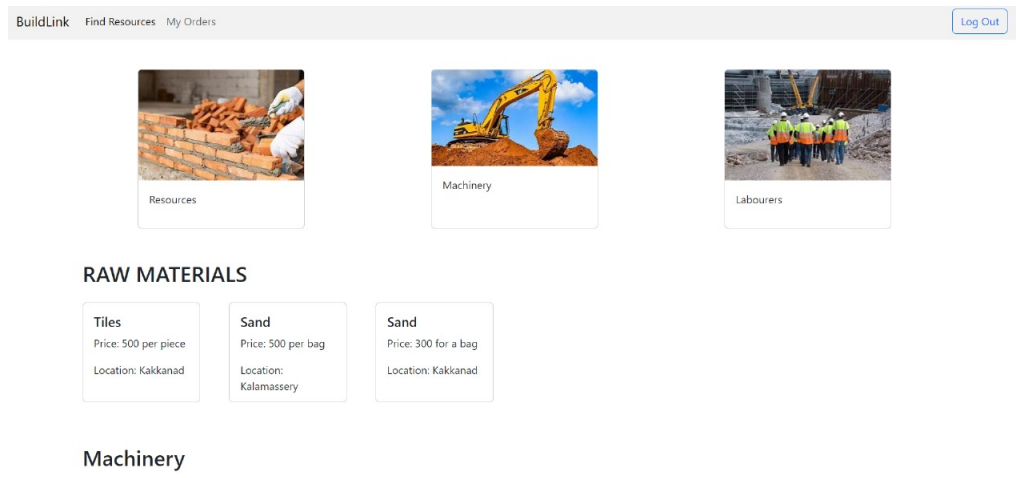



Figure 6.8: user dashboard

BuildLink Find Projects My Bids My Profile My Services  Log Out

Add Services

Choose type of Service

☐ Raw Material

☐ Machinery

☐ Labourers

Name

Price

Enter Your Location

Description

Figure 6.9: Add services page - Agent

BuildLink-Admin Panel Services All Users All Agents Effin Joe Wilson Log Out

Unverified Agents

Full Name	Email	Phone	Company Name	Address	State	District	Actions
Chris	chrisharrywilson@gmail.com						<input type="button" value="Download Image"/> <input type="button" value="Accept"/> <input type="button" value="Reject"/>
Melbin	melbin@gmail.com	0123456789	ABCD PVT LTD	Nellikunnel house	Kerala	Ernakulam	<input type="button" value="Download Image"/> <input type="button" value="Accept"/> <input type="button" value="Reject"/>
Tovin	tovin@gmail.com						<input type="button" value="Download Image"/> <input type="button" value="Accept"/> <input type="button" value="Reject"/>

Verified Agents

Full Name	Email	Phone	Company Name	Address	State	District	Actions
Effin	effinjoewilson@gmail.com	0123456789	ABCD PVT LTD	123 ABCD street	Kerala	Ernakulam	<input type="button" value="Download Image"/>

Figure 6.10: verify agents - admin


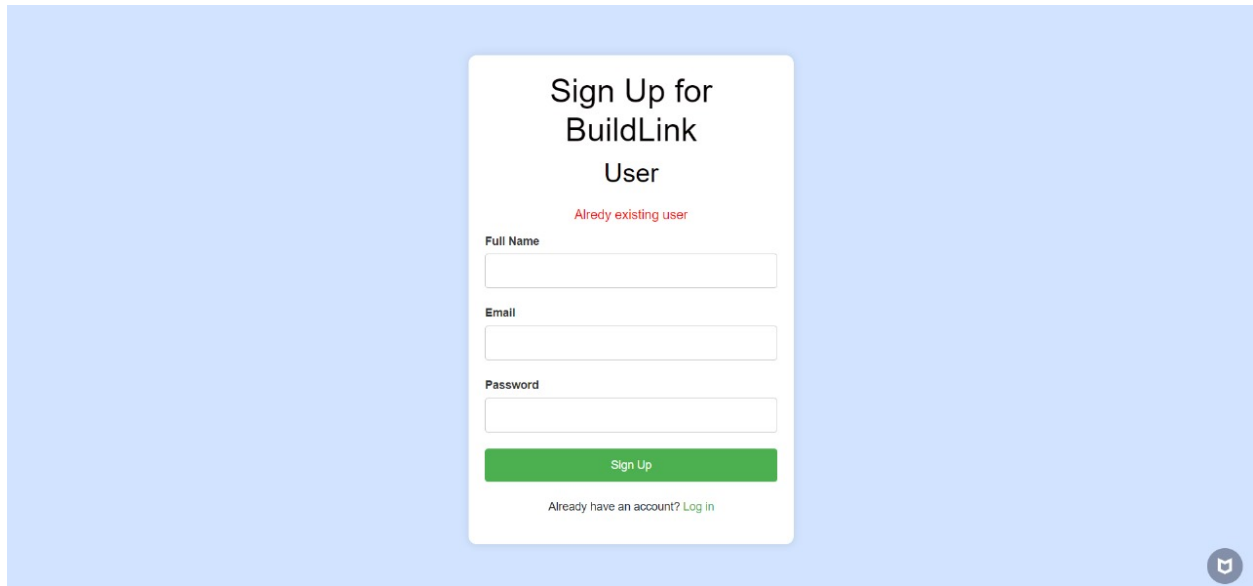
BuildLink Find Projects My Bids My Profile My Services				 Log Out
AGENT LISTING				Add Services
RAW MATERIALS				
SERVICES	PRICE	LOCATION	ACTION	
Sand	500 per bag	Kalamassery	Delete	Edit
MACHINERY				
SERVICES	PRICE	LOCATION	ACTION	
Crane	700 for 1 day	Kalamassery	Delete	Edit
LABOURS				
SERVICES	PRICE	LOCATION	ACTION	
Field Worker	700	Edappally	Delete	Edit

Figure 6.11: Agent Listing

6.3 Test Case

USER SIGNING UP

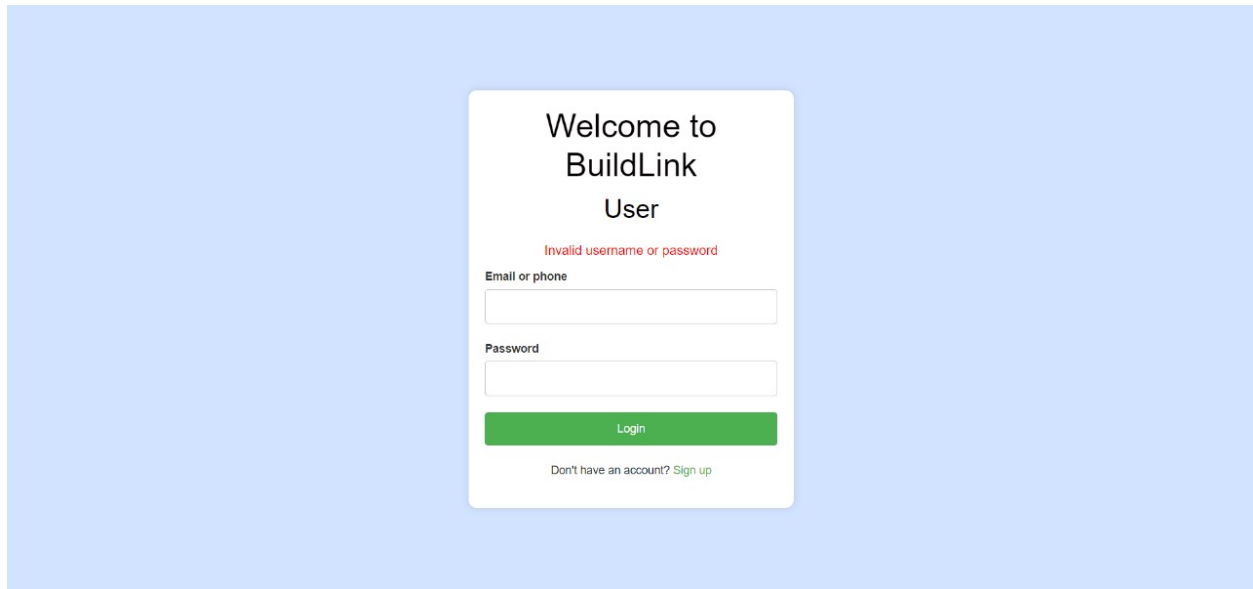


The screenshot displays a sign-up form for 'BuildLink User' centered on a light blue background. The form is a white card with the title 'Sign Up for BuildLink User'. Below the title, a red error message 'Already existing user' is centered. The form contains three input fields: 'Full Name', 'Email', and 'Password', each with a label above it. A green 'Sign Up' button is positioned below the password field. At the bottom of the form, there is a link that says 'Already have an account? Log in'. In the bottom right corner of the light blue background, there is a small circular icon with a white 'B' on a dark blue background.

Figure 6.12: Test Case 1

In the signup page the error condition checks if `SignupErr` exists in the session. If true, it displays an error message indicating "Already existing user" centered below the input fields, informing users that the provided email is already associated with an existing account during the signup process.

USER LOGIN



The image shows a user login form for 'BuildLink' centered on a light blue background. The form is a white card with a shadow. It contains the following elements: the title 'Welcome to BuildLink User', an error message 'Invalid username or password' in red text, two input fields labeled 'Email or phone' and 'Password', a green 'Login' button, and a link 'Don't have an account? Sign up'.

Welcome to
BuildLink
User

Invalid username or password

Email or phone

Password

Login

Don't have an account? [Sign up](#)

Figure 6.13: Test Case 2

In the login page the error condition checks if `loginErr` exists in the session. If true, it displays an error message indicating "Invalid username or password" centered below the input fields, alerting users to incorrect login attempts.

Chapter 7

Result

The project succeeded in creating BuildLink, a web-based platform revolutionizing the construction industry. It provides a one-stop platform for Users seeking construction resources and Agents offering these services to connect, contribute, and collaborate efficiently. Access to BuildLink is restricted to verified industry professionals, ensuring a secure and focused environment.

Successfully able to register new Users and Agents and sign in. Users can search for construction resources such as raw materials, machinery, and skilled labor through an advanced search page. Agents can list their services, including detailed descriptions and pricing. Users and Agents can add posts such as project updates, resource availability, and industry news on the home page. Profiles for every user are automatically updated with their listed services, interests, and posts shared by them. Users can search for other users using the search button, view their profiles, and connect with them. Can sign out using the sign-out button.

Chapter 8

Future Scope

1. Messaging Features:

Develop robust messaging features, including direct messaging, group chats, and secure document sharing within the chat interface. This will improve communication between Users and Agents, facilitating quicker decision-making and more effective collaboration.

2.Transaction Features:

Implement secure transaction features, allowing Users to request quotes, place orders, and make payments directly through the platform. This will streamline the procurement process, making it more efficient and secure for both Users and Agents.

3. Expansion to Include Industry Networks:

Expand the platform to include a network of industry professionals, allowing Users to connect with experts, mentors, and potential business partners. This feature can provide valuable networking opportunities and support professional growth and collaboration.

4. Virtual Events and Webinars:

Develop capabilities for hosting virtual events, webinars, and live-streamed sessions. This will allow the construction community to engage in real-time interactions, attend industry conferences, and participate in training sessions from anywhere in the world.

5. Mobile App Enhancements:

Continuously improve and update the mobile app version of BuildLink, adding new features, optimizing performance, and ensuring a seamless user experience across different devices. This will enable Users and Agents to manage their activities on the go.

6. Accessibility and Inclusivity:

Focus on improving accessibility features to ensure that users with disabilities can access and navigate the platform effectively. Consider features such as screen reader compatibility, color contrast options, and keyboard navigation to create an inclusive environment for all users.

Chapter 9

Conclusion

BuildLink - A web-based platform revolutionizing the construction industry has made it simple for Users and Agents to connect and collaborate efficiently.

Now, the difficulties of managing construction resources and finding reliable service providers are no longer an issue. It is highly useful for all industry professionals to find resources, connect with service providers, and communicate seamlessly. All project updates, resource availability, and important industry news are readily available to every user. BuildLink stands out as a focused platform for the construction industry, providing a straightforward solution without any extrinsic issues.

Chapter 10

References

Books :

1. Fundamentals of software engineering, Rajib Mall, Pearson Education, 2011. ISBN: 978-8131761624
2. Learning Node.js Development” by Andrew Mead, Packt Publishing, 2018. ISBN: 978-1788626854
3. MongoDB: The Definitive Guide” by Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, O’Reilly Media, 2019. ISBN: 978-1491954461

WebLinks :

- 1.<https://styled-components.com/docs/basics>
- 2.<https://nodejs.org/en/docs/>
- 3.<https://expressjs.com/en/starter/installing.html>
- 4.<https://docs.mongodb.com/>
- 5.<https://github.com/ericf/express-handlebars>
- 6.MDN Web Docs for JavaScript](<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)