A REPORT

ON

VARIOUS TASKS

DEVELOPED DURING

**“Web App Internship”**

Submitted to

**ETI LABS**



UNDER THE GUIDANCE OF

**Mr. Rachit Thukral**

**Acknowledgements**

I am profoundly grateful to Mr. Rachit Thukral for his expert guidance and continuous encouragement throughout the project tasks and also want to thank Mr. SRN Reddy and Ms. Manasi Mishra and Mr. Rachit Thukral for giving this opportunity.

* **Saurav Kumar**

**TASK-1**

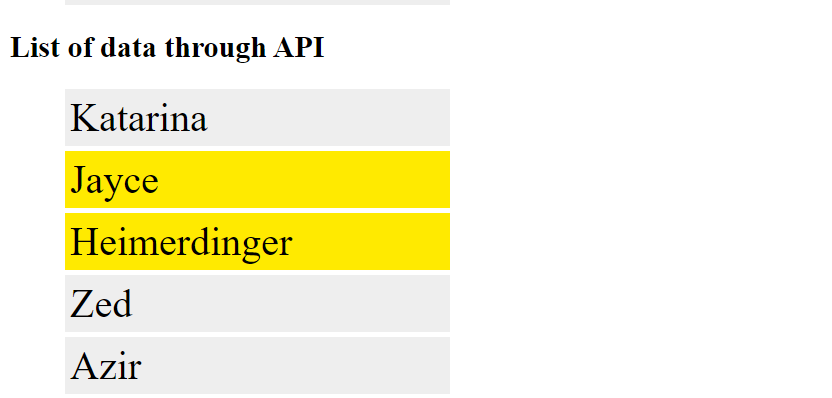
**AIM:** Create a hello world Web App to get familiarize with the Handlebars framework.

**Technologies:** Node.js, Express-handlebars, html, css

**Input:** API Data

**Output:**





**TASK-2**

**AIM:** Explore features of the handlebars and use them in single Web App

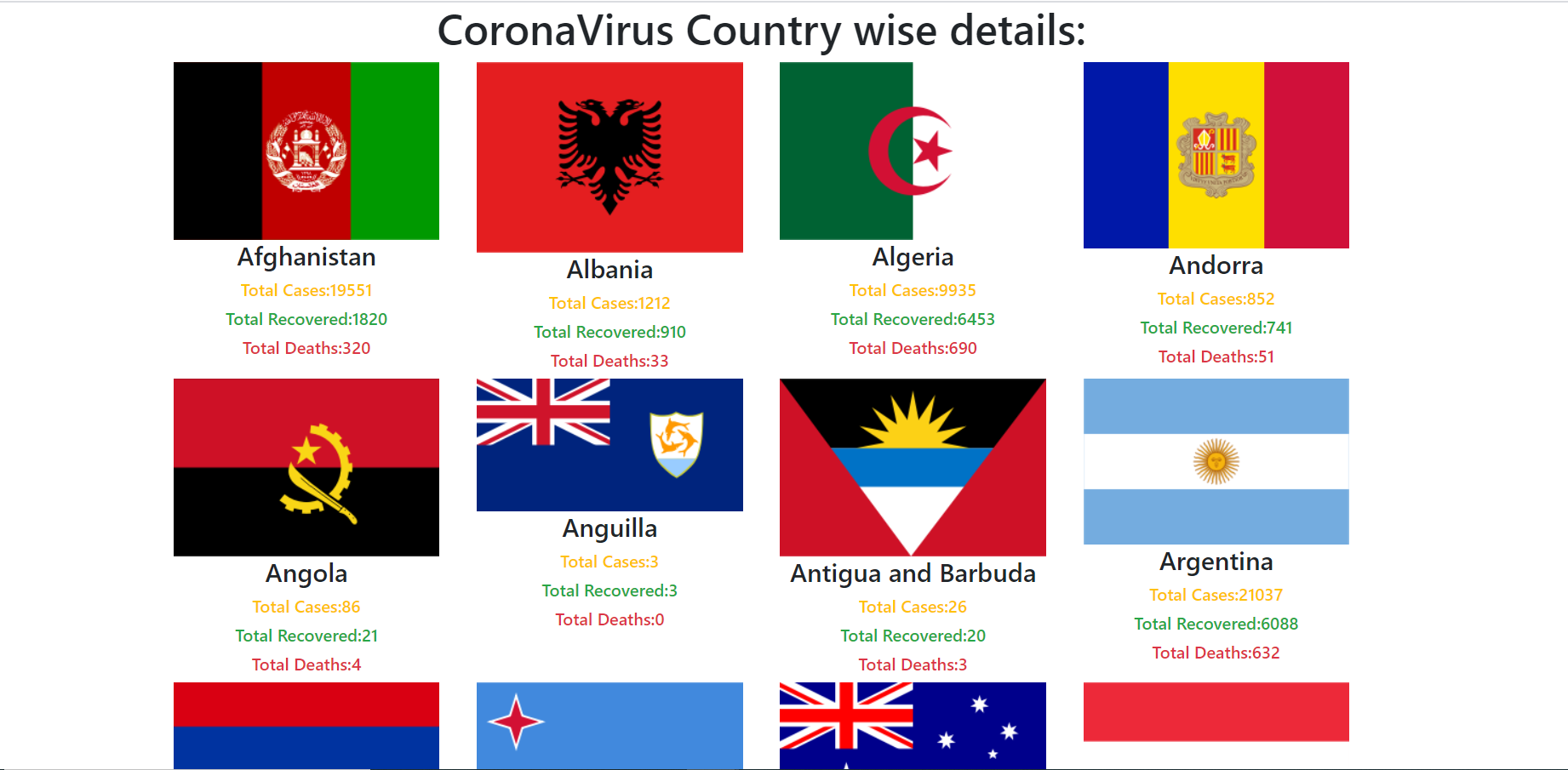
**Technologies:** Node.js, Handlebars.js, html, css

**Input:** Novelcovid.json data file

**Project:** listing out corona cases country wise details.

**Handlebars functions used**: Basic Expressions, Built-in helpers like loops,conditions,custom-helpers, sub-expression, partials

**Output:**



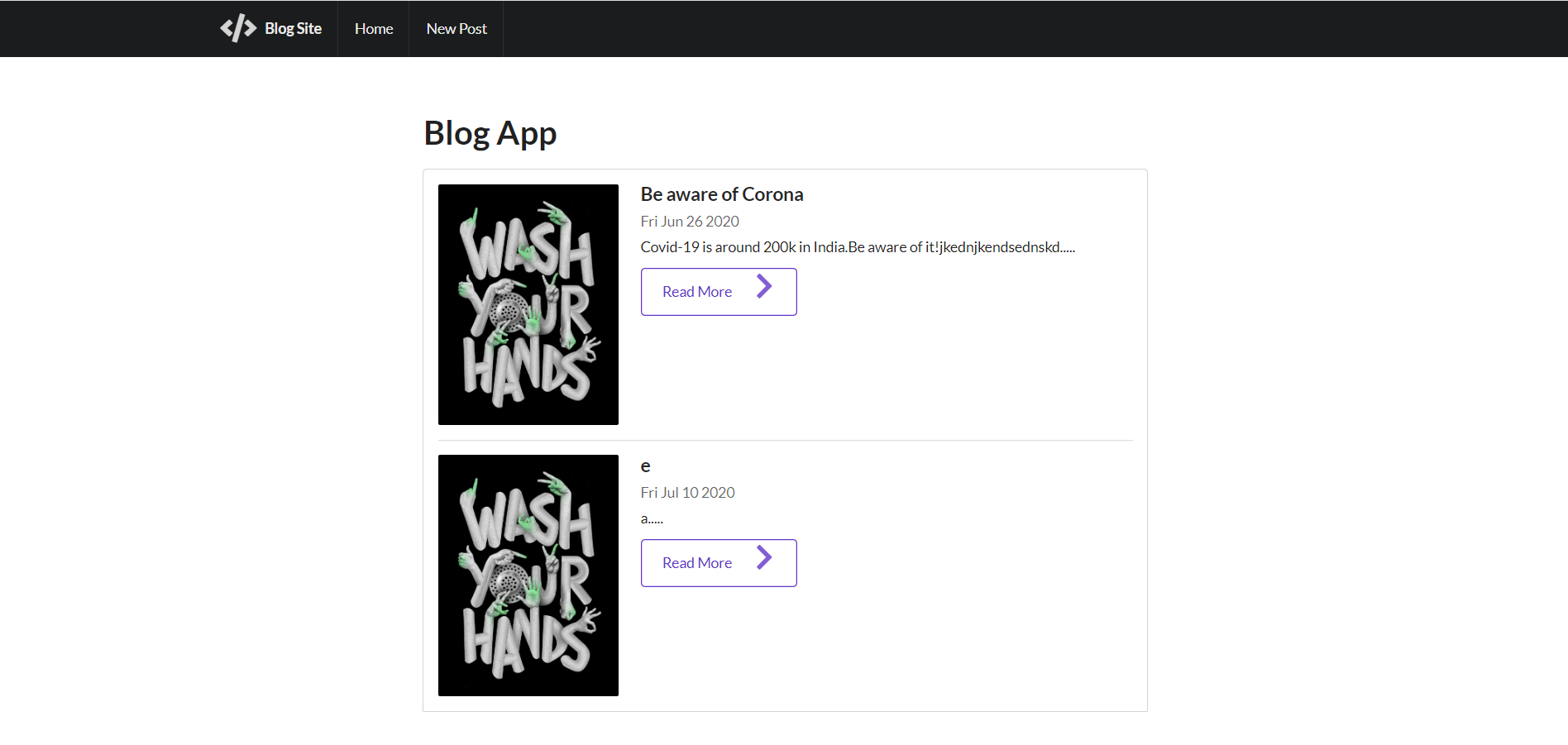
**TASK-3**

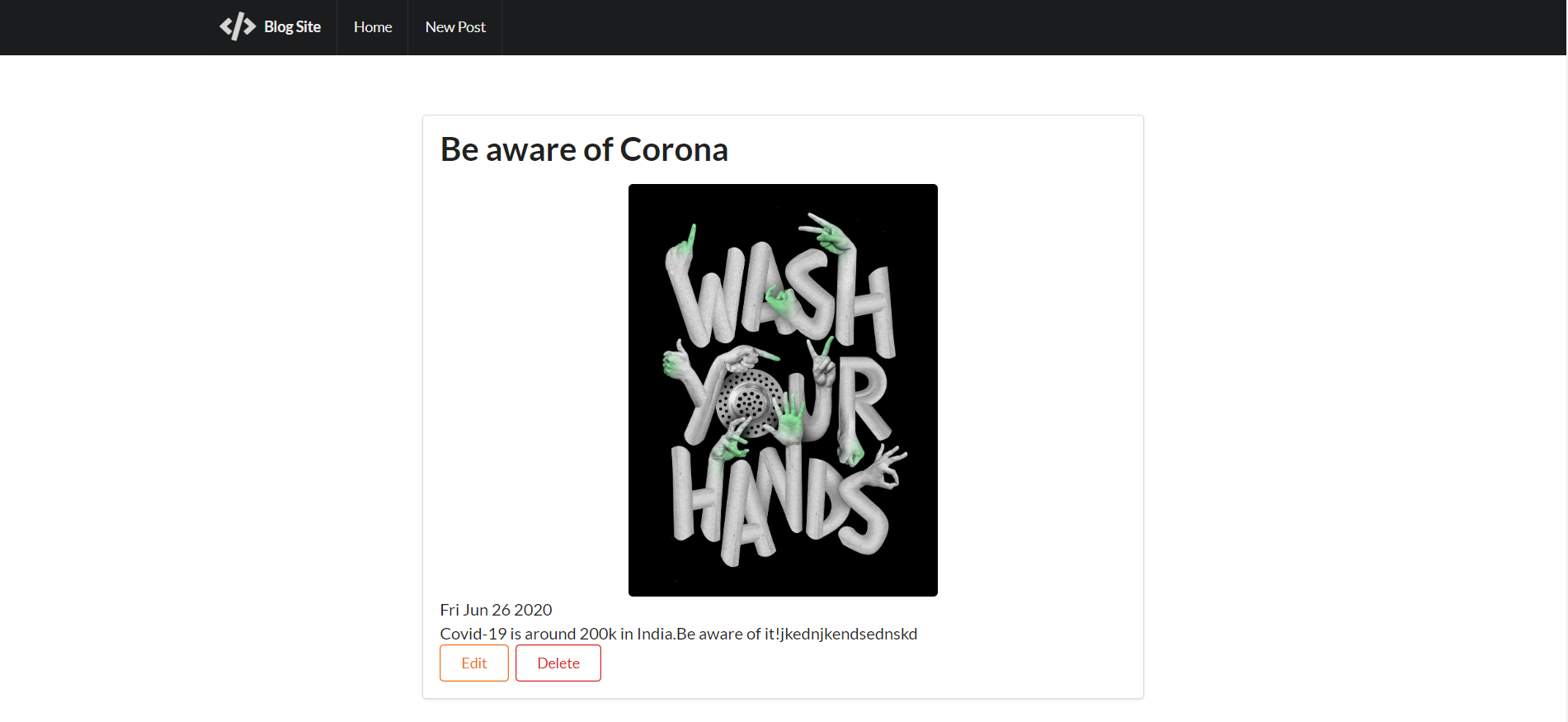
**AIM:** Make a project of your choice using mongoDB

**Description: Read/Write in MongoDB database, Display data from database, Add/edit/del data from front end**

**Technologies:** Node.js, express, MongoDB, ejs, html, css

**Project:** Public Blog App where one can upload, delete, edit blogs in the dashboard.

**Output:**



**TASK-4**

**AIM:** Explore ways for making responsive and reusable components and use in web app

**Description: List below mentioned and make a summary document for making responsive and reusable components:**

**1. CSS functions**

**2. JavaScript methods (front end)**

**3. Handlebars template**

**A component will be combination of all the above and use partial for that. Make node.js functions to call the components with parameters to set various properties of the components like background color and passing data to it.**

**Example - Image gallery component**

**input data - images**

**properties - background color, gap between images etc.**

**Technologies:** Node.js, handlebars.js, html, css

**Input:** images url, width between images

**Output:**

**TASK-5**

**AIM:** Make a reusable and responsive header bar which accepts three lists as input

**Description: Input - 3 lists (list item can be image and/or text)**

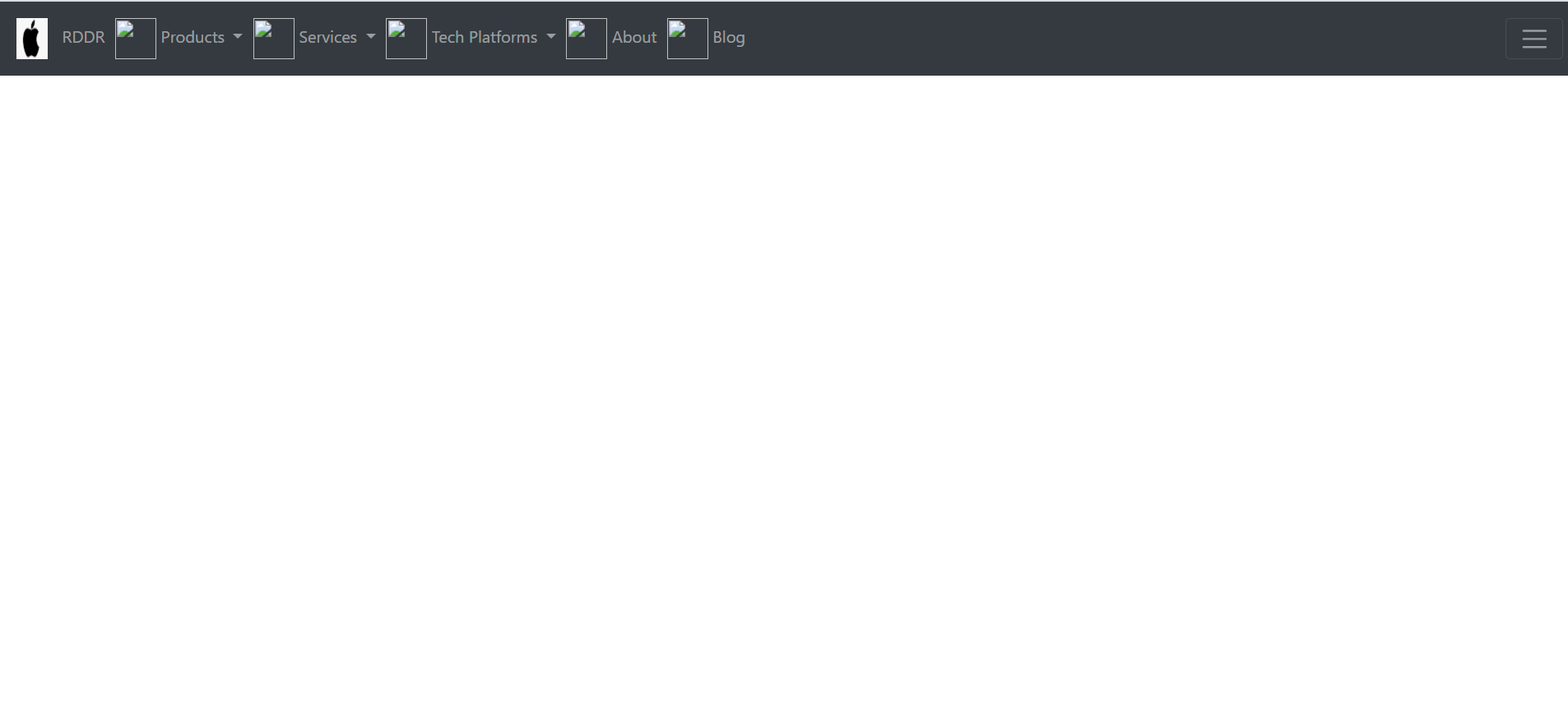
**On laptop screen - list 1 & list 2 visible on header, list 3 as drop down menu (like 3 lines icon for more)**

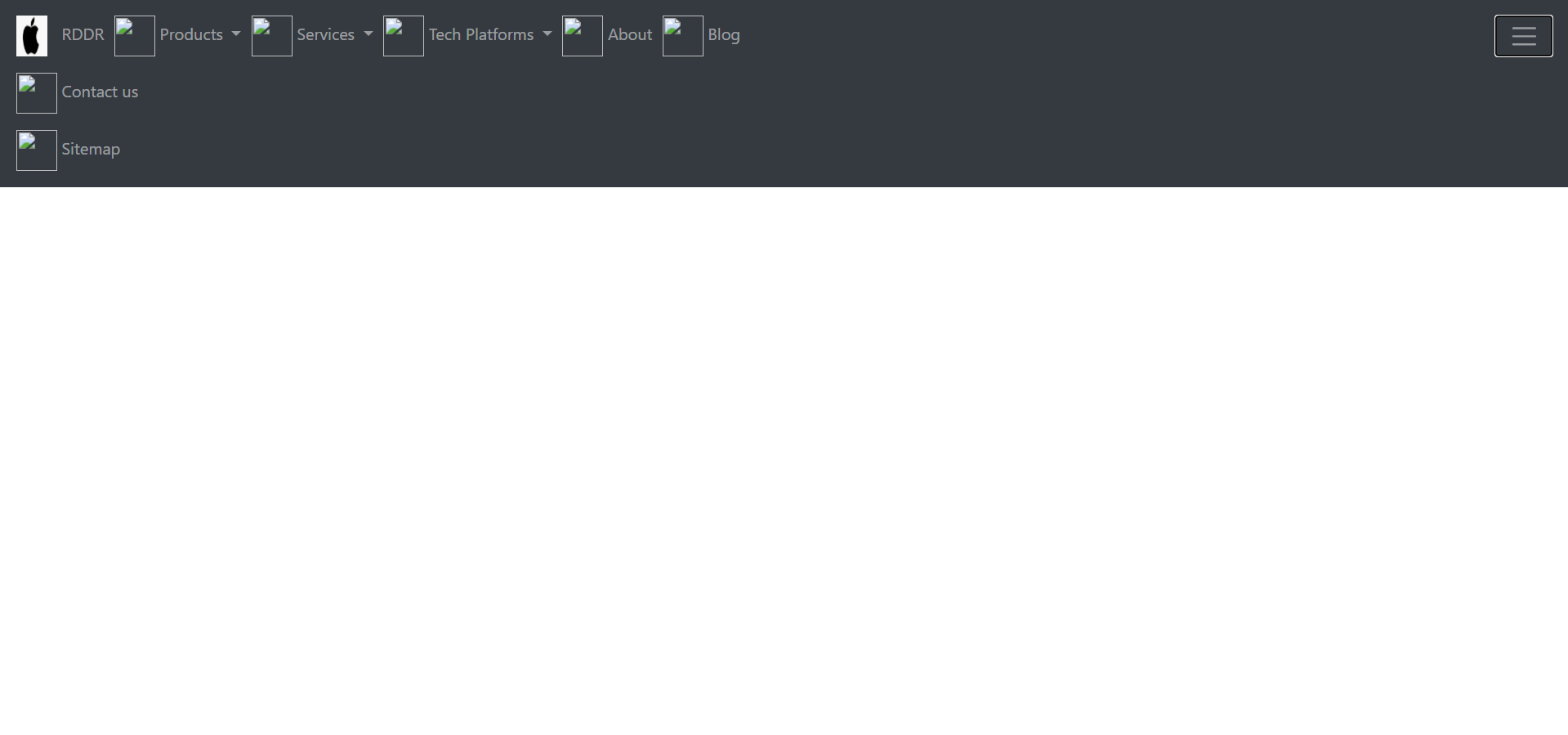
**On mobile screen - list 1 visible on header, list2 & list 3 as drop down menu**

**Technologies:** Node.js, handlebars.js, html, css

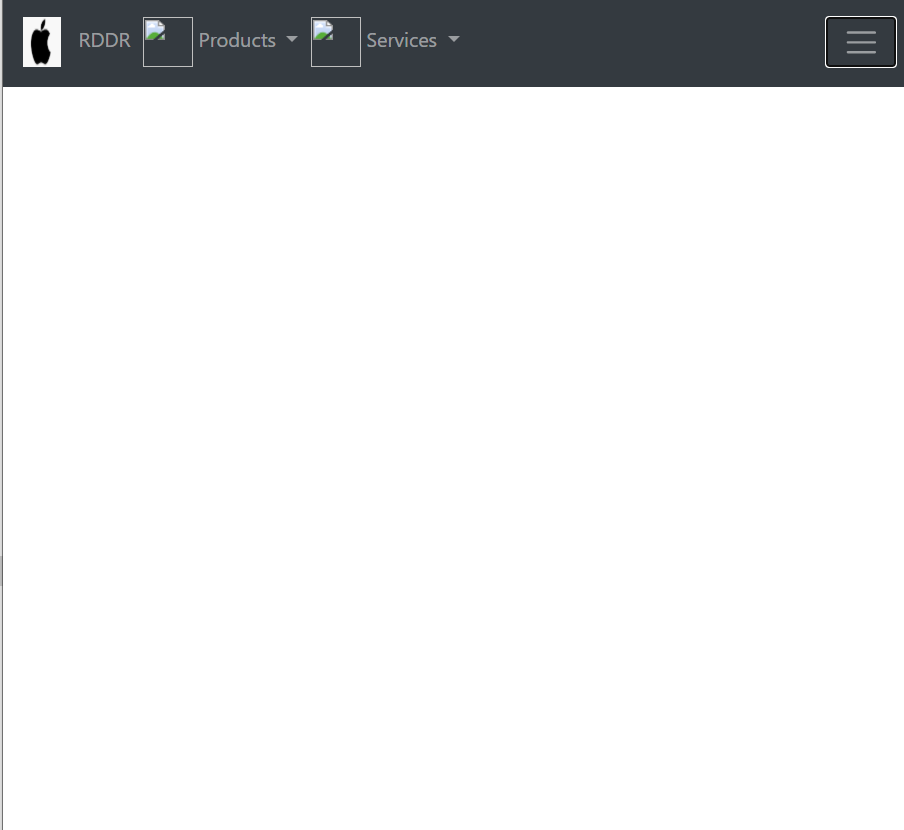
**Input:** images or text list or both(link or dropdown)

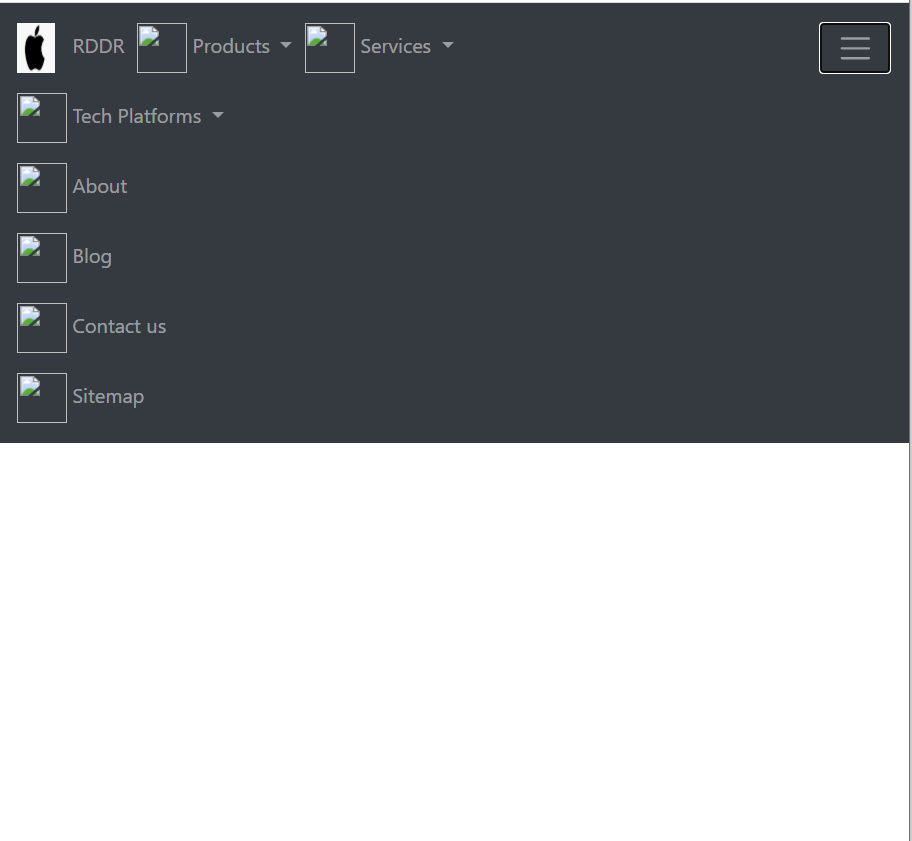
**Routes:** Main- /five, Embedded- /five/products/esd, IoT- /five/products/iot, Trainings- /five/services/trainings, IIoT- /five/services/iiot, Code editor- /five/techplatforms/codeeditor, Protocols- /five/techplatforms/protocols, About- /five/about, Blogs- /fve/blogs, Contact us- /five/contactus, Sitemap- /five/sitemap.

**Output: Laptop screen**



**Mobile screen**





**TASK-6**

**AIM:** Make sidebar as a tree structure of page links

**Description: Use partial template**

**Input - list of links (list item cab be image and/or text)**

**On laptop screen - list should be visible by default and toggle option for show/hide**

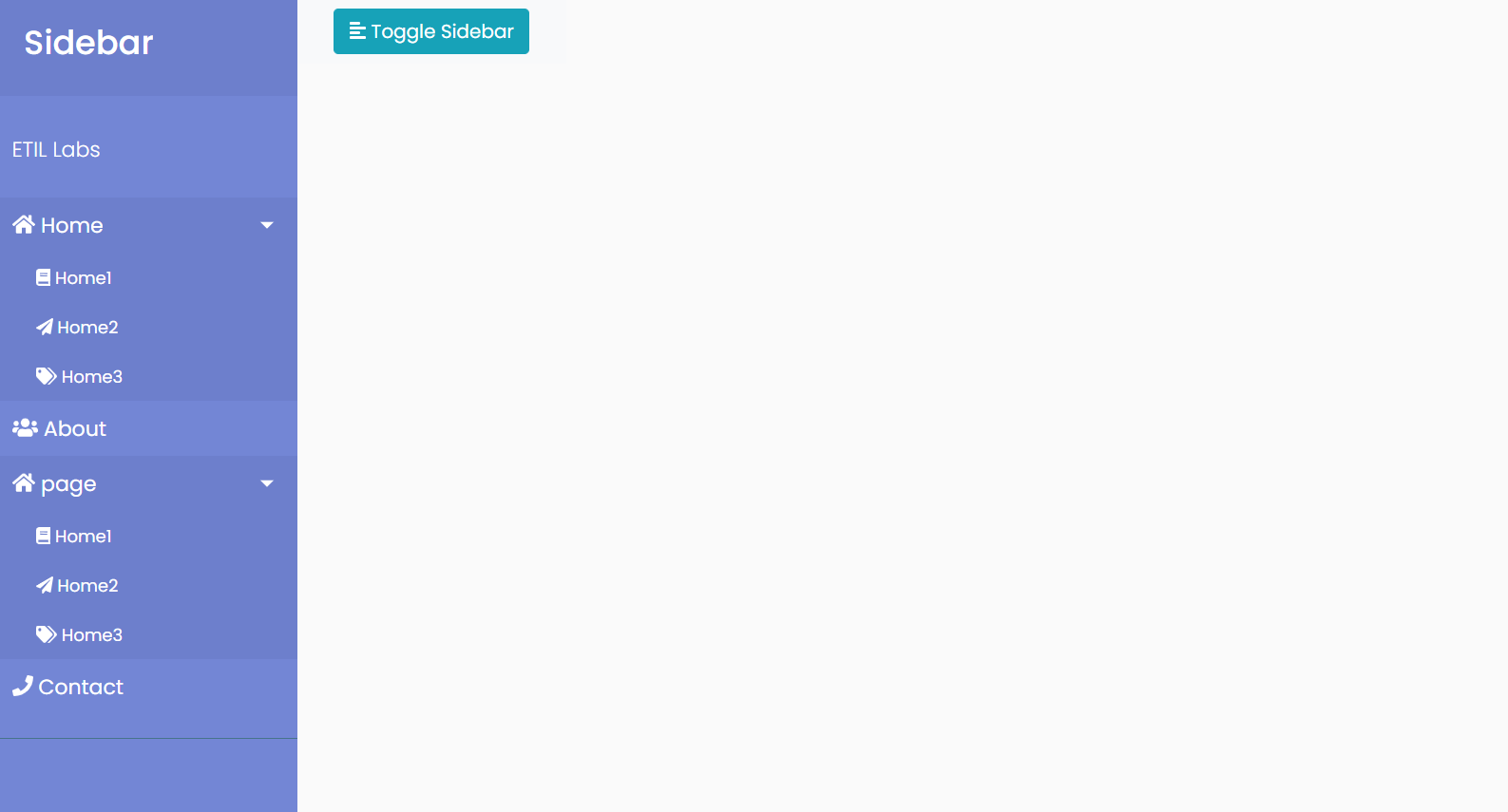
**On mobile screen - list should not be visible by default but toggle option for show/hide**

**Technologies:** Node.js, handlebars.js, html, css, js

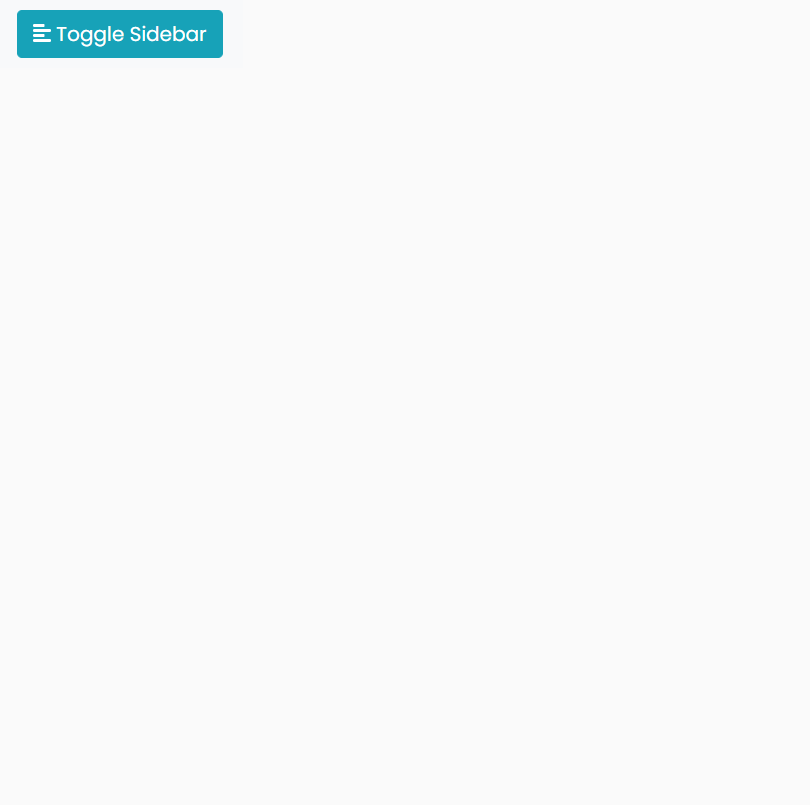
**Input:** images or text list or both(link or dropdown)

**Routes:** Home- /six, Home1- /six/home1, Home2- /six/home2, Home3- /six/home3, About- /six/about Contact - /six/contact

**Output: Laptop screen**



**Mobile screen**



**TASK-7**

**AIM:** Make a web app where you can use all the components you made or will made

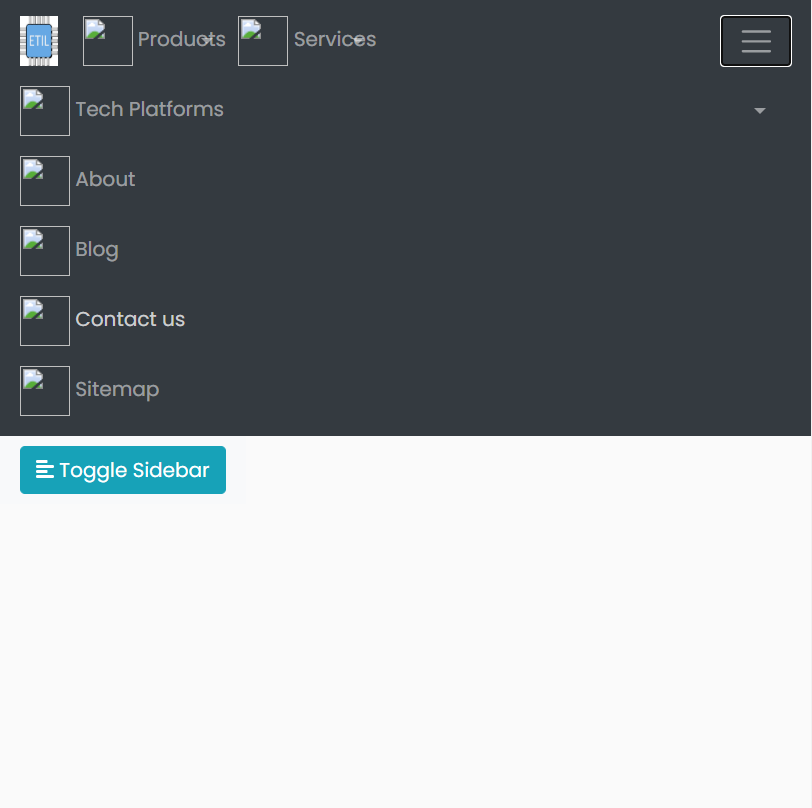
**Description:** MergeTask 5 and Task 6

**Technologies:** Node.js, handlebars.js, html, css, js

**Output:**

**Laptop screen:**

**Mobile screen:**



**TASK-8**

**AIM:** Image/Video slide show component

**Description: Component input data - images/videos path, heading & description for each image/animated image/video**

**Upon clicking it will pop out and there will option for choosing the image/video:**

**1. Video can be played with heading & description displayed**

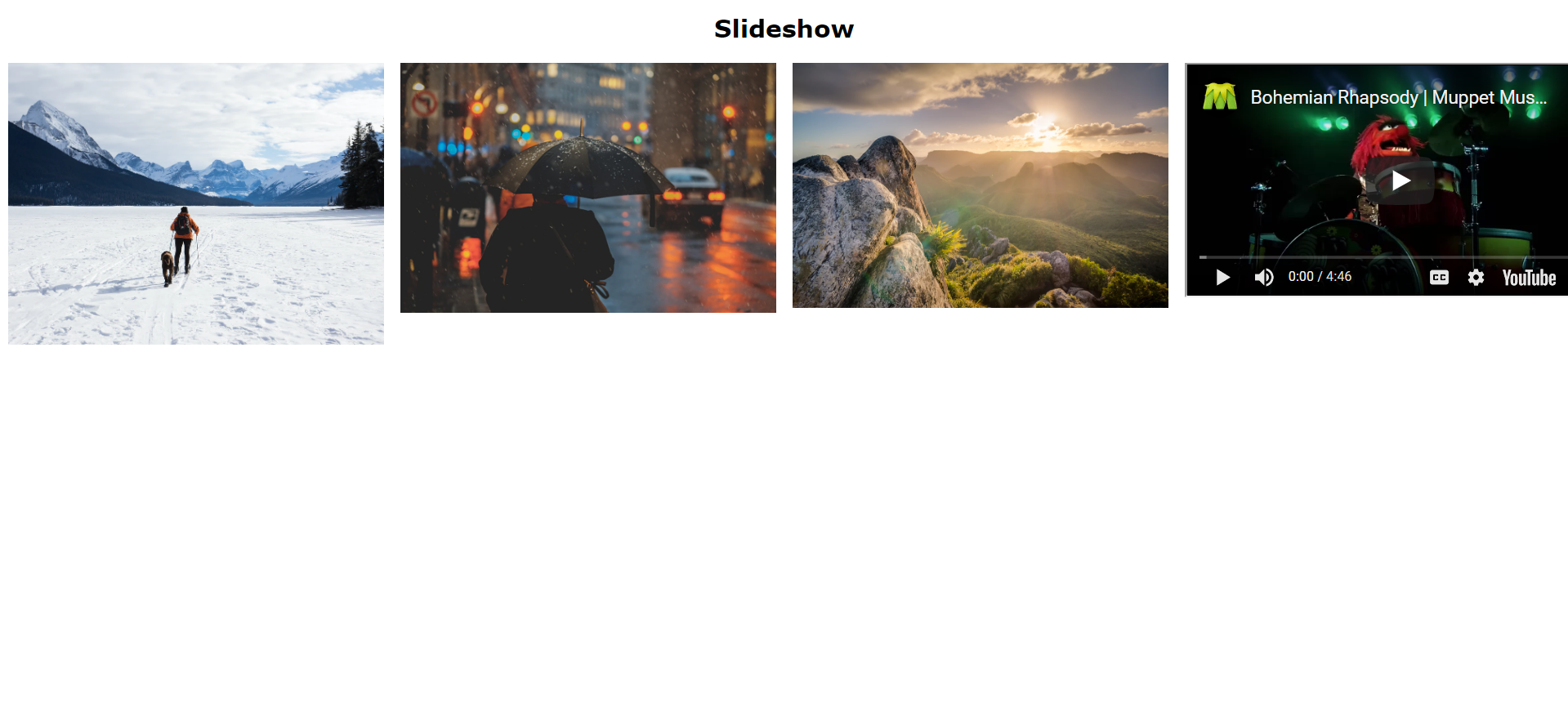
**2. Image can be viewed with heading & description displayed**

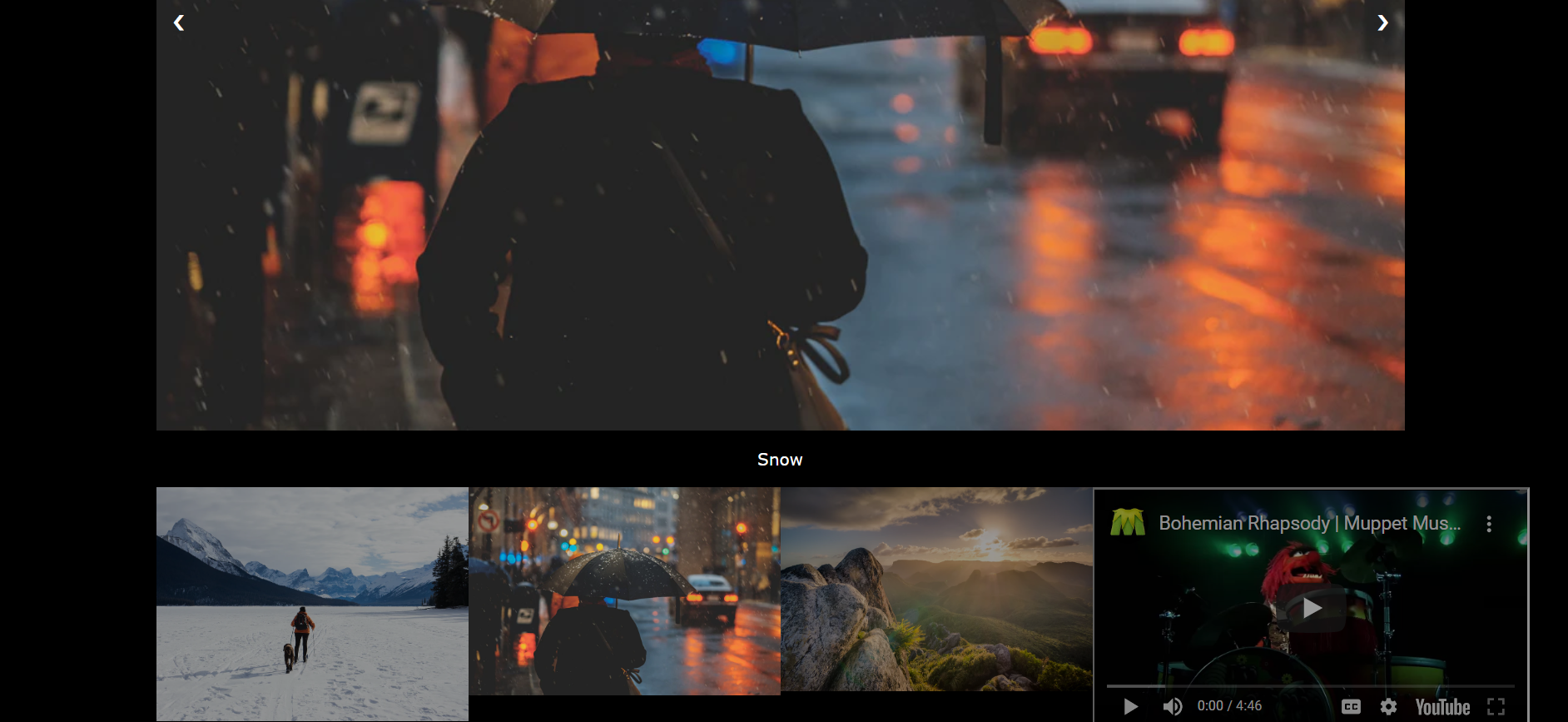
**3. Animated image should animate with heading & description displayed**

**Technologies:** Node.js, handlebars.js, html, css, js

**Output:**

**Normal:**

**POP-OUT Feature:**



**TASK-9**

**AIM:** Make product components

**Description: Component 1: preview of products**

**UI should look like** [**http://www.etilabs.com/shop/**](http://www.etilabs.com/shop/)

**Component 2: detail of product**

**UI should look like** [**http://www.etilabs.com/product/eti-labs-embedded-system-development-board-8051/**](http://www.etilabs.com/product/eti-labs-embedded-system-development-board-8051/)

**Store products information in database (mongodb)**

**Make pages to use these components**

**Technologies:** Node.js, MongoDB, handlebars.js, html, css, js

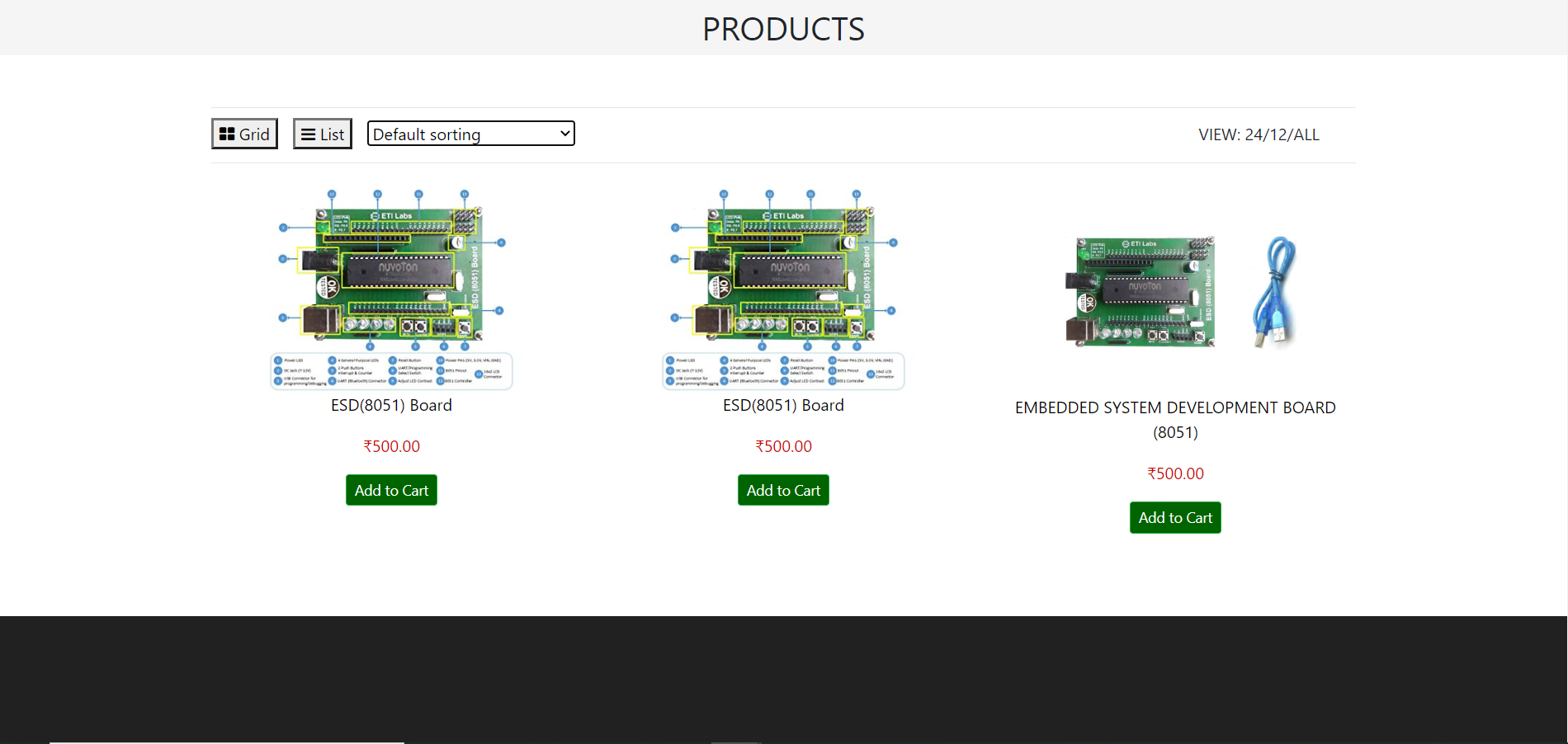
**Input:**

Component1:All pictures, names and prices of products are stored in MongoDB**(route url:/nine)**

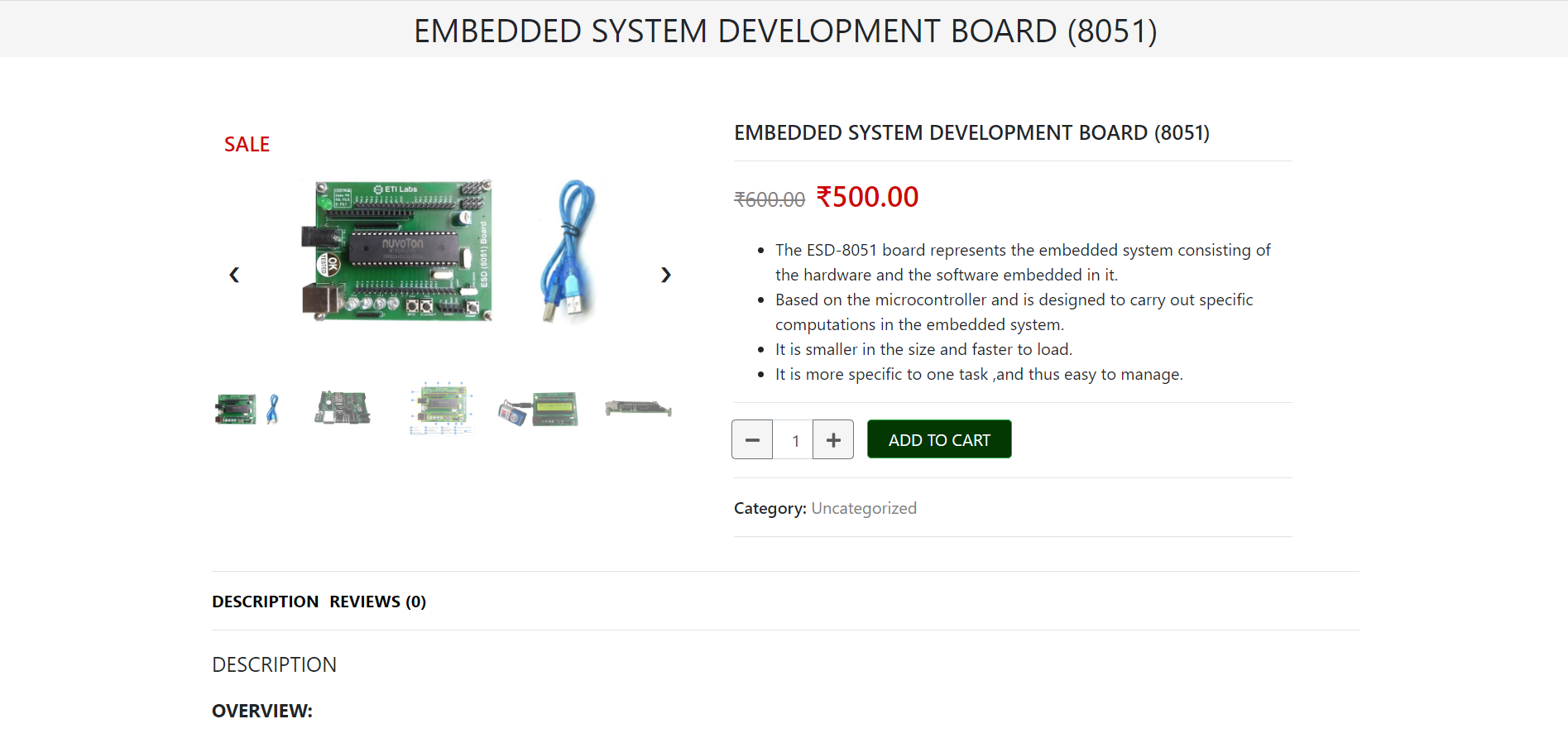
Component2: All pictures, headings, product’s prices and details, slideshow images, description details are stored in Database**(route url:/nine2)**

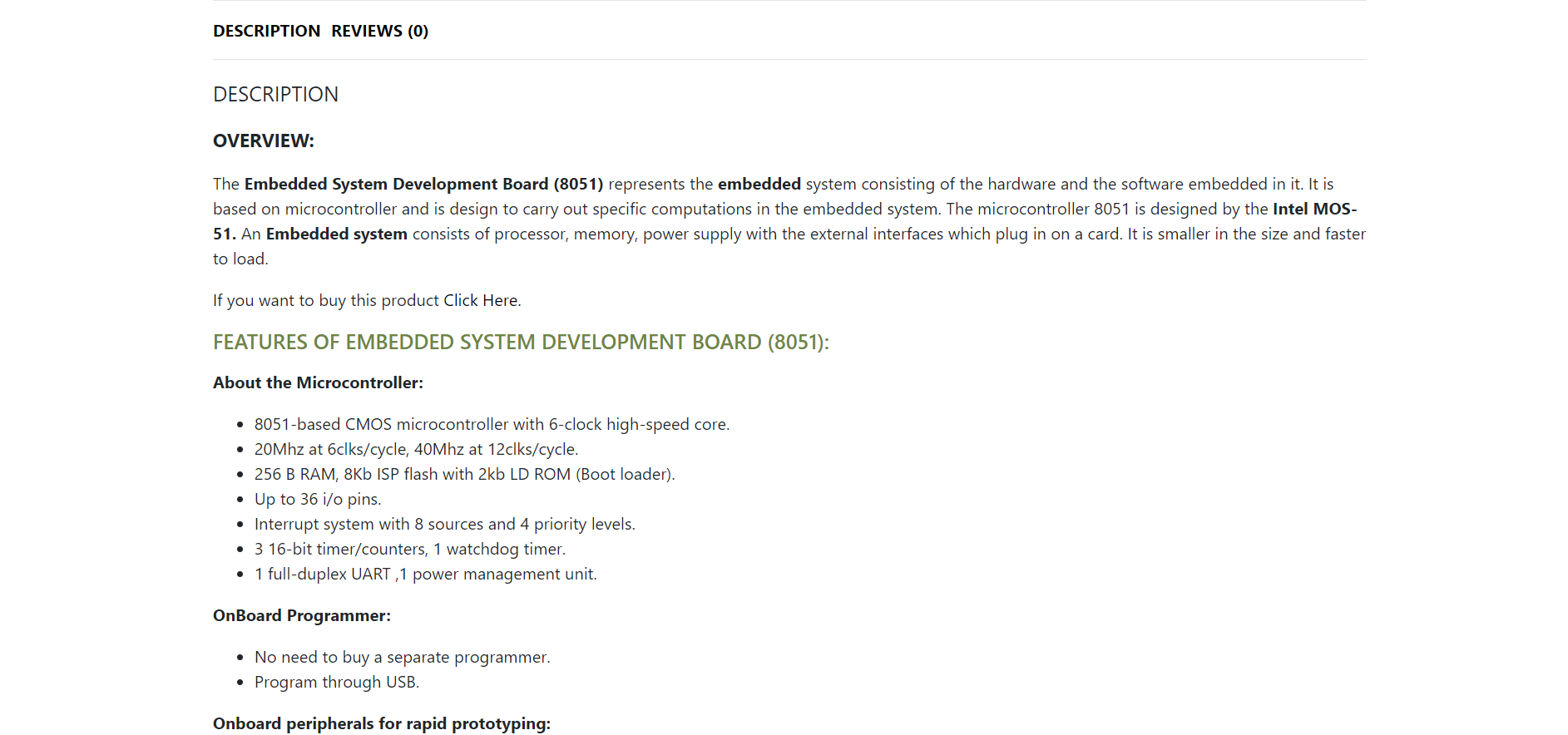
**Output:**

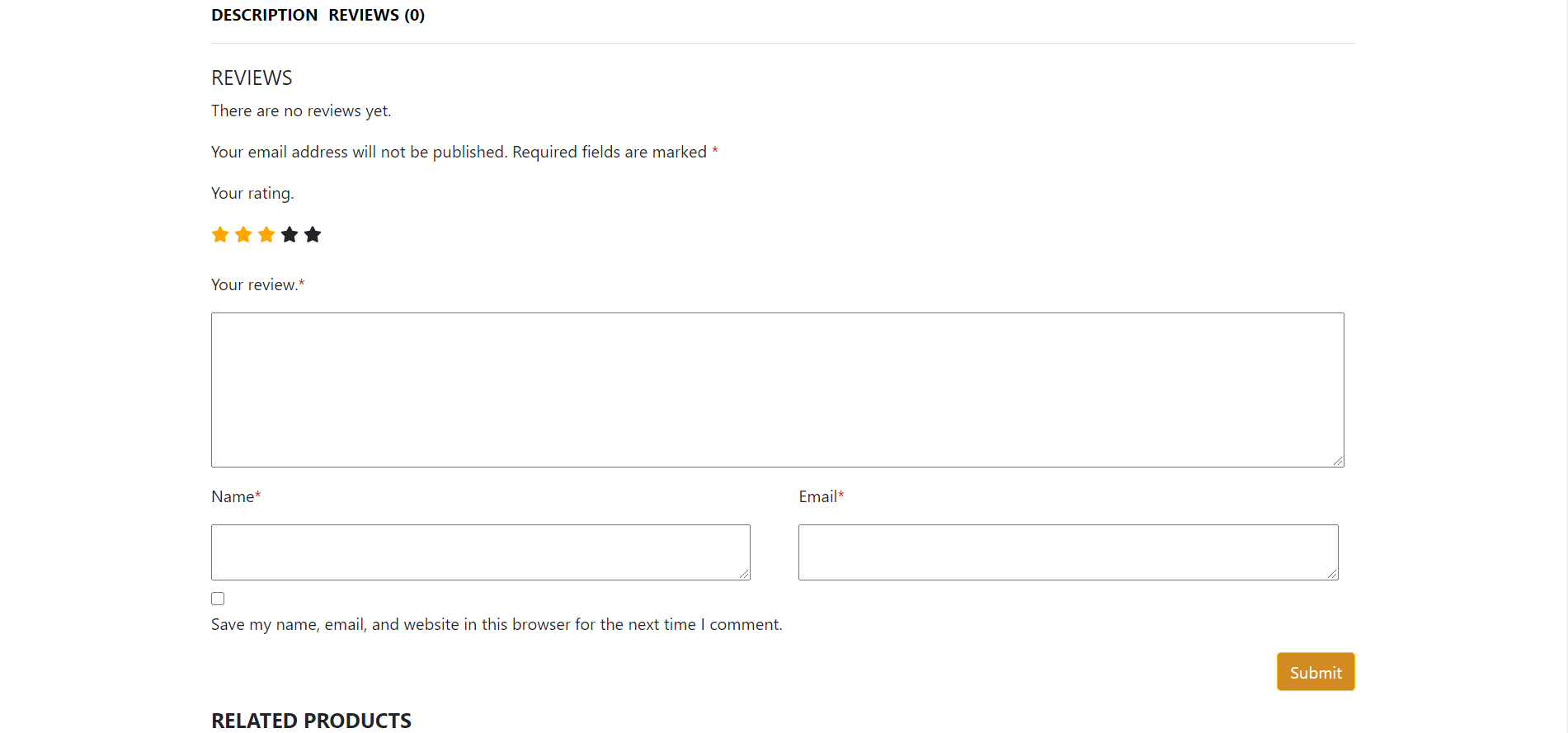
**Component 1:**

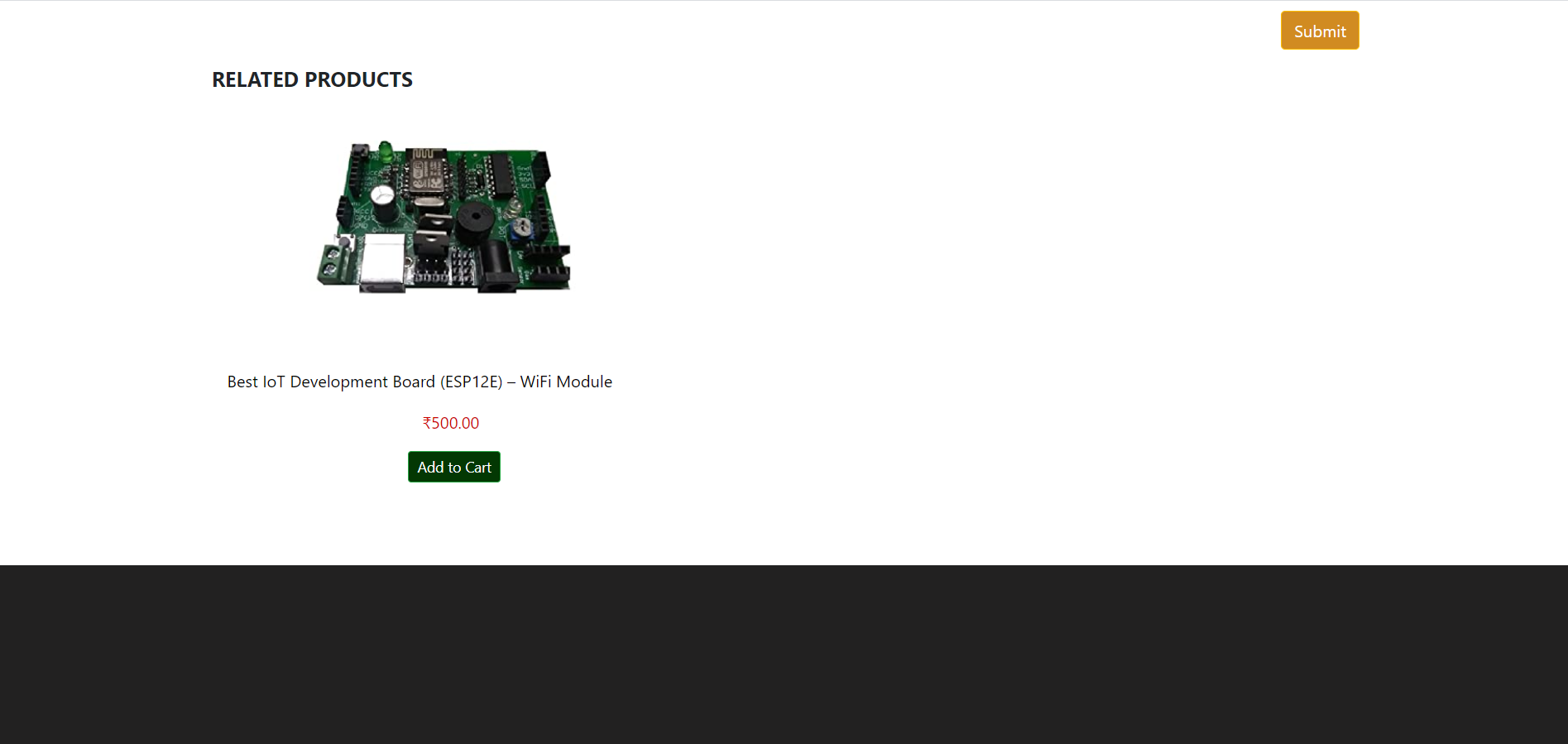


**Component 2:** Both description and reviews toggle upon clicking.









**TASK-10**

**AIM:** Make user authentication (signup, signin, roles of user) using mongodb

**Description: Component 1 - sign up**

**Component 2 - sign in**

**Roles of user: Admin, Normal**

**[will update more roles later]**

**Database must store username, password (encrypted), role**

**When a user requests a URL then at server side we should know if its signed in or not. If signed in, which user so that only pages with permissions to specific users will be served.**

**Make 3 pages for testing roles:**

1. **only admin can view**
2. **admin and normal user can view**
3. **anyone can view (even without signup)**

**Technologies:** Node.js, MongoDB, handlebars.js, html, css, js

**Input:**

Home Page: **(route url:/ten)**

Login Page:**(route url:/ten/login)**

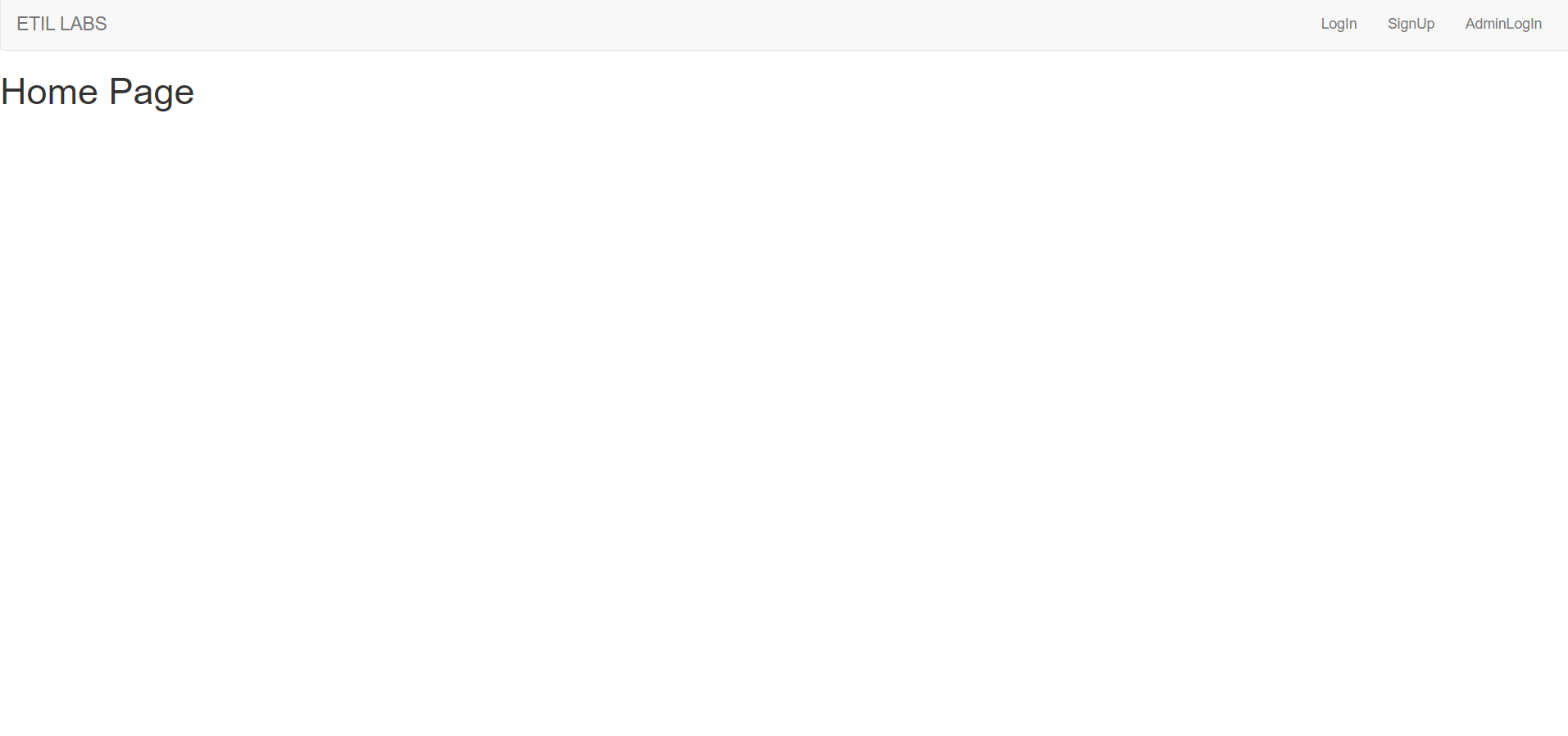
SignUp Page:**(route url:/ten/register)**

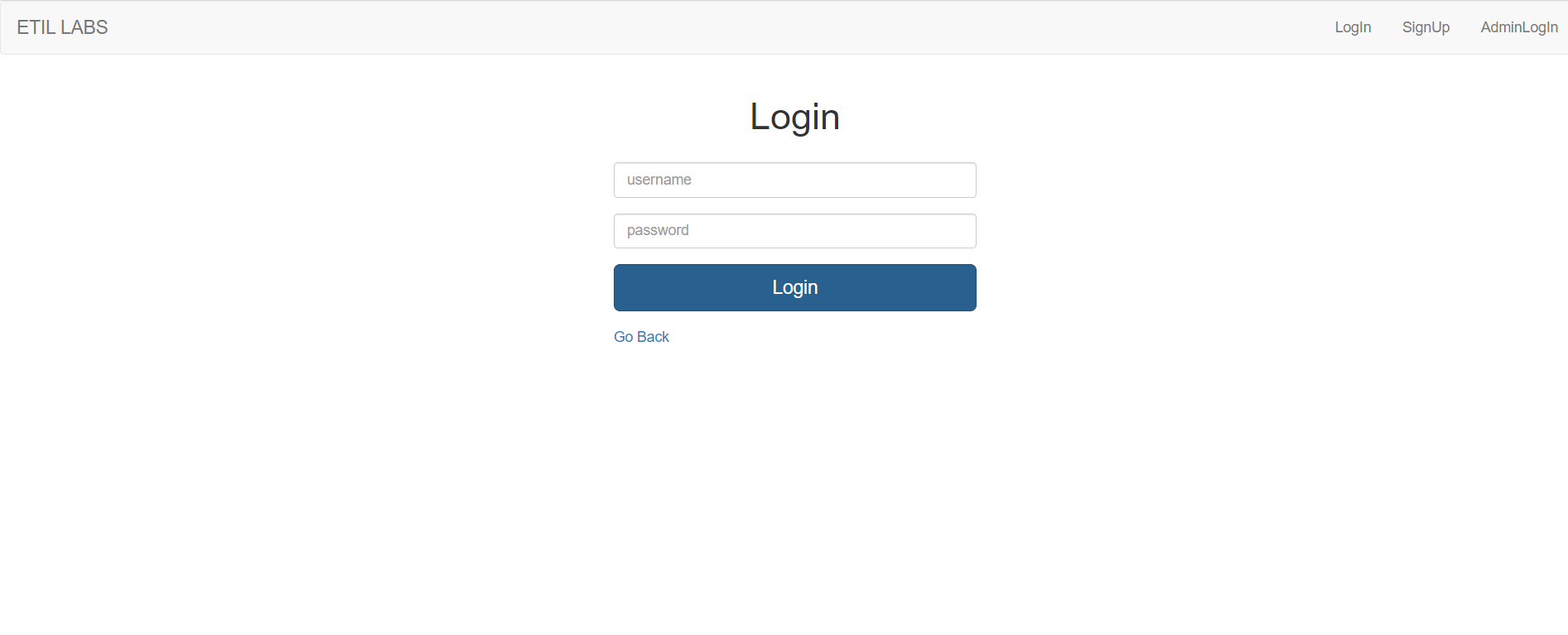
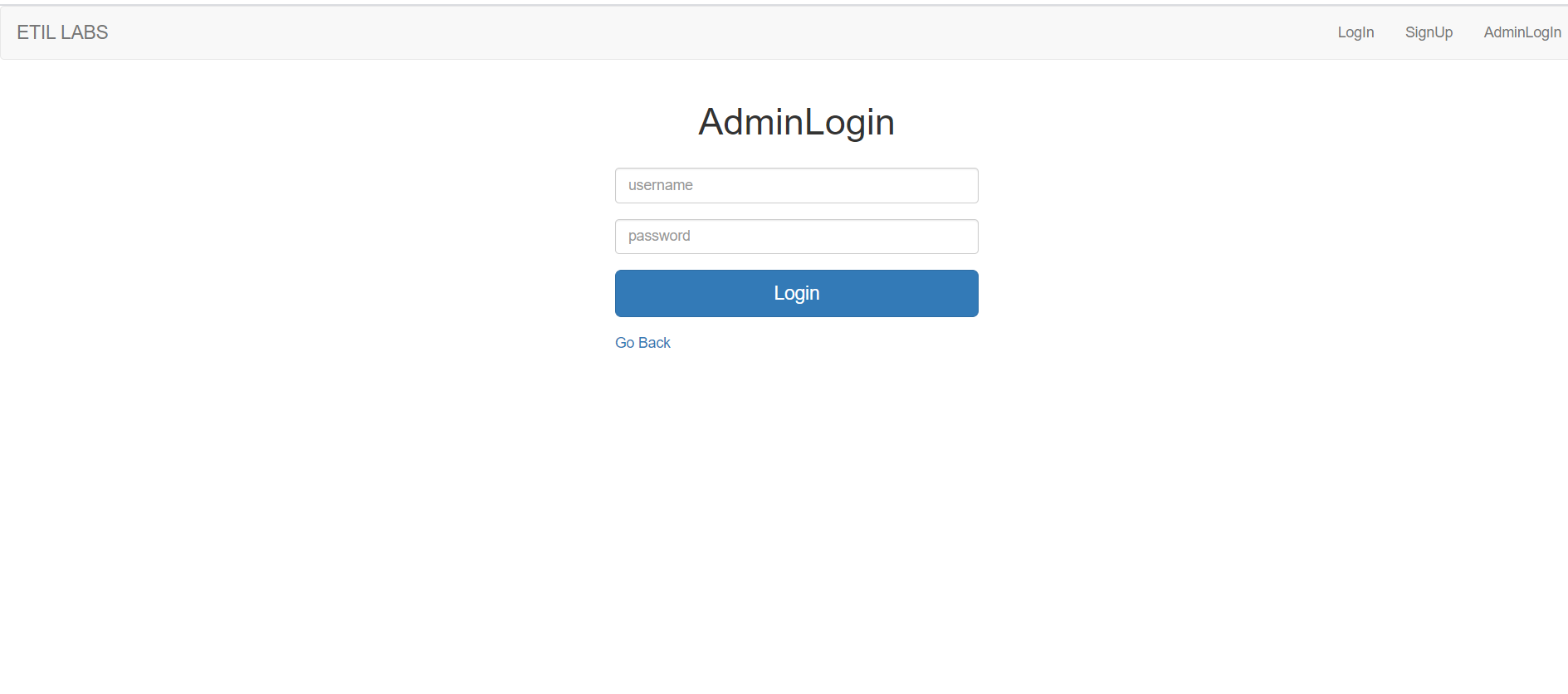
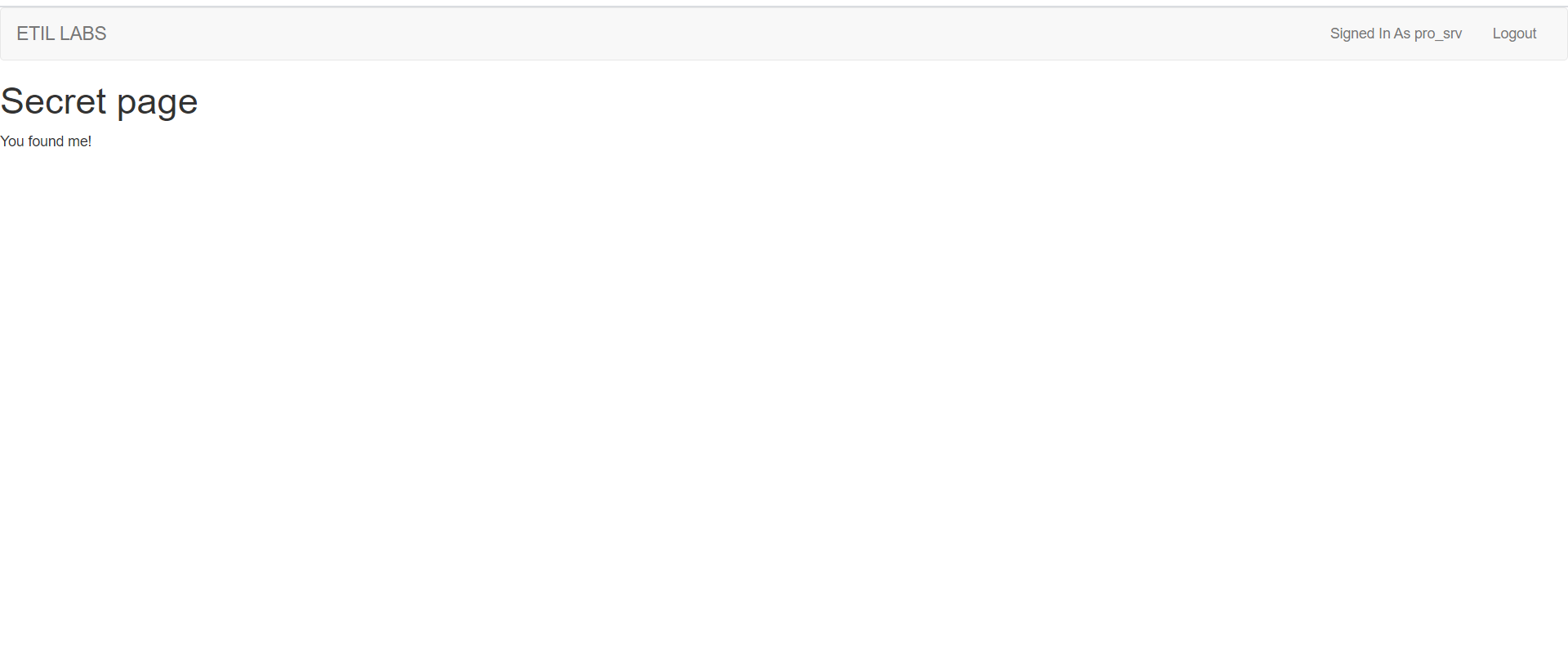
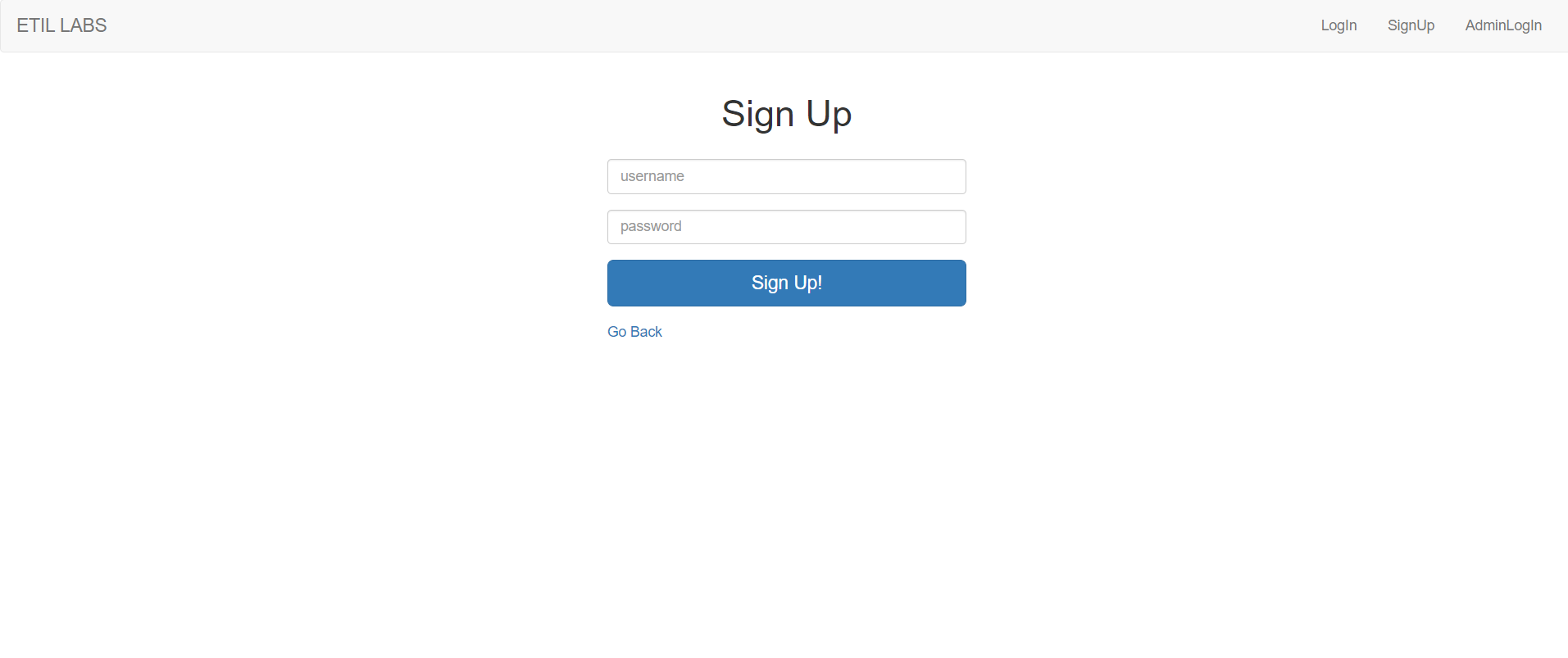
AdminLogin Page:**(route url:/ten/adminlogin)**

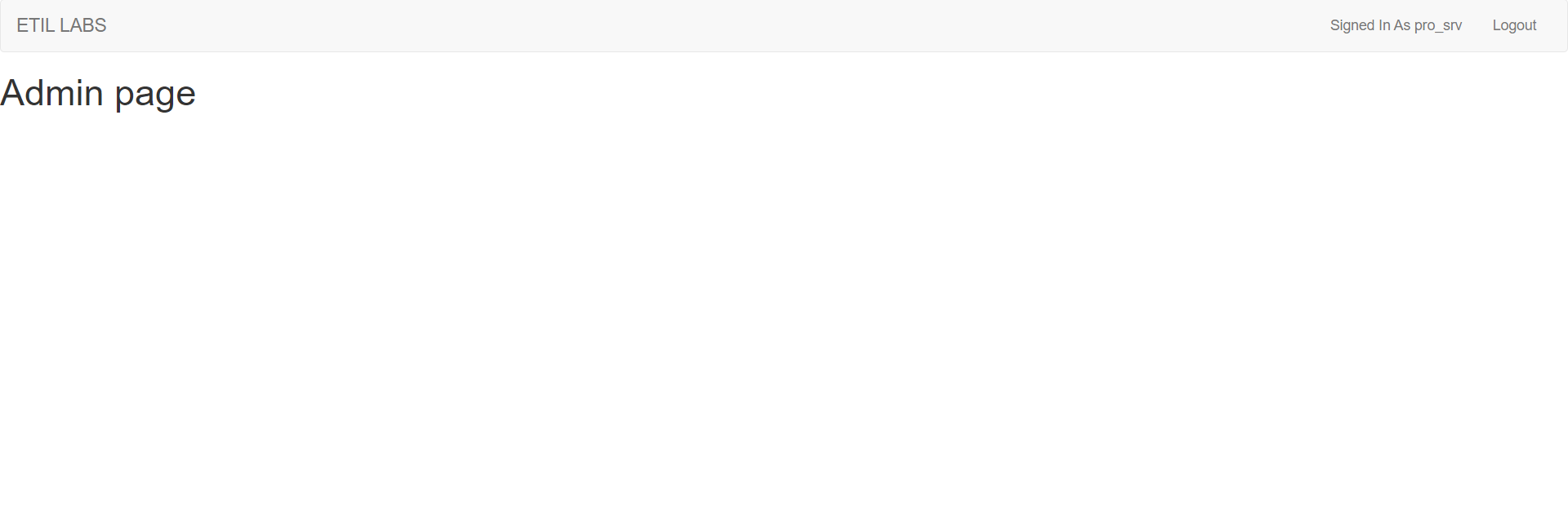
Secret Page:**(route url:/ten/secret)**

Admin Page:**(route url:/ten/admin)**

**Features**

* Password is encrypted and decrypted using crypto module
* Session is maintained with cookies
* After logging out cookies are deleted
* Username, Ip address and password is stored in database
* After logging in, username is displayed on the navbar 





**TASK-11**

**AIM:** Make a node.js app to read and write cookies on client-side (browser) for the user

**Technologies:** Node.js, handlebars.js

**Output:**

Multiple cookies are created and deleted

