# **Project : Simply Blogging**

Course-end Project 1

DESCRIPTION

As a Full Stack Developer, you have to develop a blog for the company where people can post articles and customers can benefit from the content.

**Background of the problem statement:**

As the project is progressing, management has asked you to create a blog page for the company website so that newsletters and other informative articles can be posted by experts for the benefit of the customers.

**You must use the following:**

* Visual Studio Code
* Bootstrap
* HTML/CSS
* Javascript
* GitHub

Date : 14/04/2023

# **Project Name : Project1\_SL\_Blog.**

# **Brief: Create a blog page for the company website**

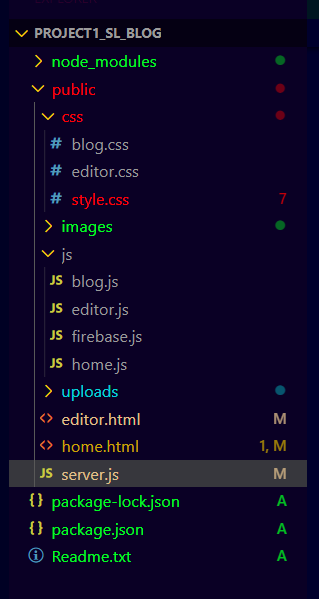
# **1. Code:**

# This is based on Node.js. Install NPM and Node.js in order to start.

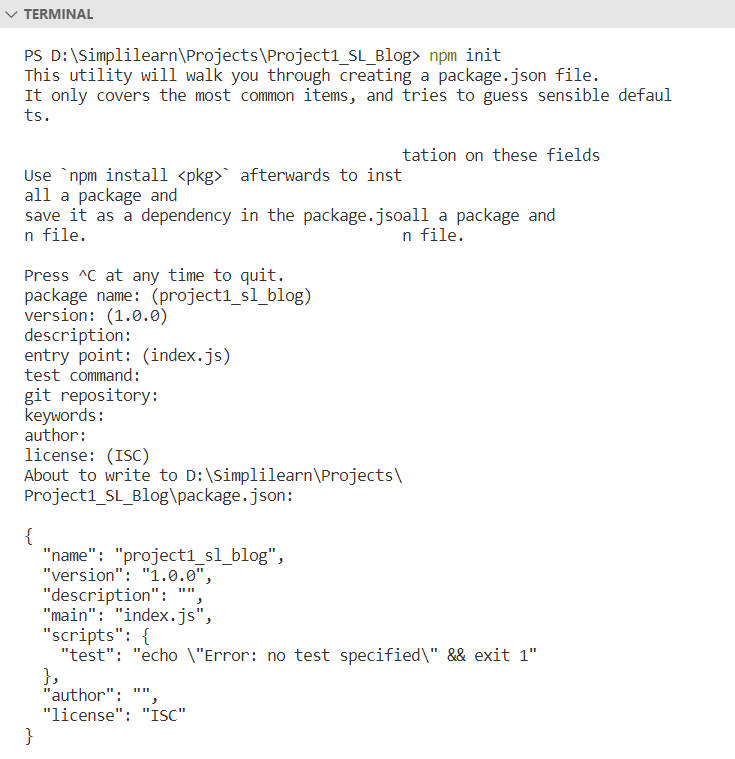
## **Create a New Project in Visual Studio Code**

Make a Folder …\Project1\_SL\_Blog.

## **Folder Structure**



## **Initialize npm**



# **2. Server**

Open the project file (root directory) in your code editor. Open the Terminal and run

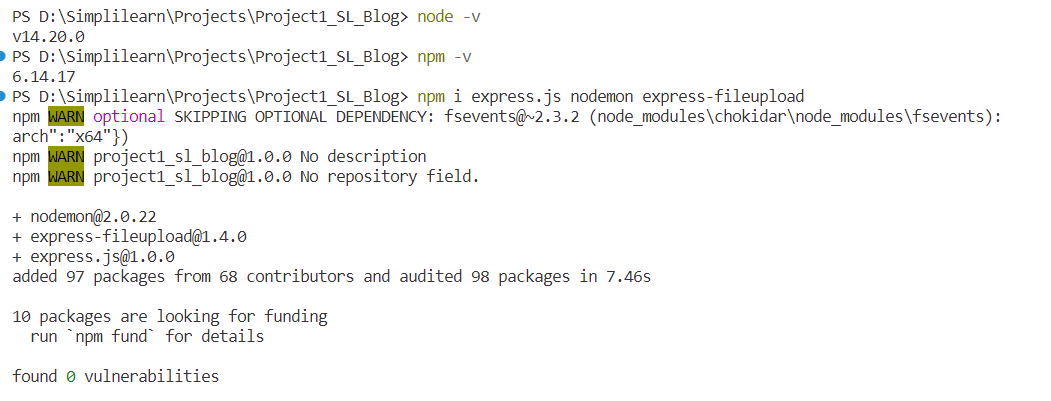
**npm init**

This will initialize NPM to our project. After that install some packages by this.

**Run the following command**

**$ npm i express.js express nodemon express-fileupload**

-express.js - is to create a server  
-express-fileupload - is to handle uploads  
-nodemon - is to run server continuously

once package installed. 

It will create a node\_modules folder in project directory which contains express.js ,nodemon , express-fileupload.

You should see a package.json file inside your root directory. Open it.

And change it scripts to

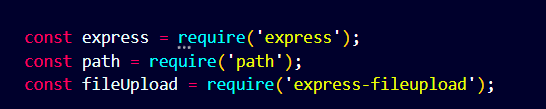


The next step is to create a server

Create a new file inside your root directory name it server.js. And open it.

First import all packages that we need.

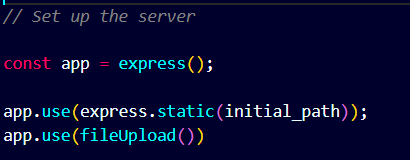
**Start by importing all packages**



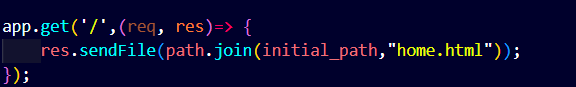


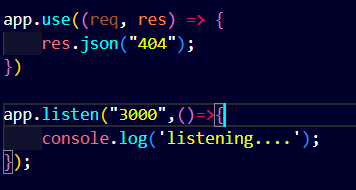
**Set up the server**

use app.use(fileupload()) to enable file uploads.



After this make a home route and in response send ***home.html*** file. And run your server on 3000 port.





Note : Port Number can be changed.

Run your server by **node server.js**

****

**Server is ready now.**

Open the browser with the port number. The following picture is just for testing



# **3. HTML Page**

As we have created two .html files

Home. html

Editor.html

**Home.html**

Write a basic HTML structure and link the home.css file. Then start by creating a navbar.

***Structure***

1. Create a Nav bar
2. Create the header
3. Create the sections using bootstrap & CSS:
   1. About us
   2. Blog Card
   3. Detailed Blog Card Sections (Images & Description).
   4. Footer

**Head**

* 1. Include fonts File
  2. Include Bootstrap.min.css
  3. Include Style.css

**HTML**

Style.css

****

**Navbar**

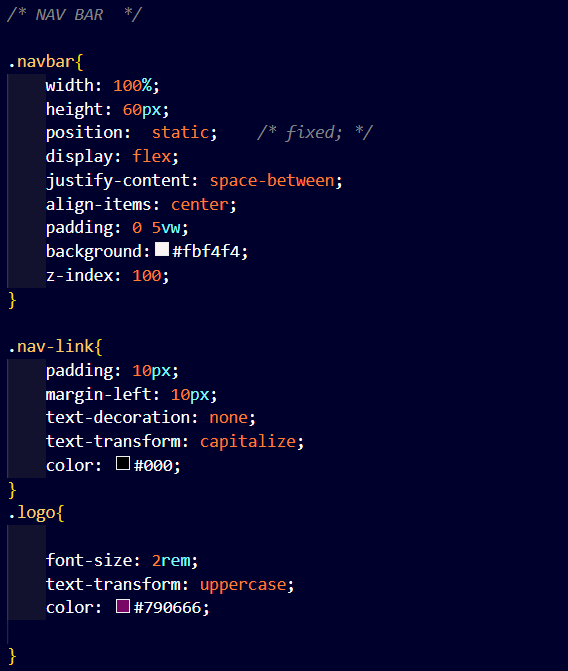
****

**Code**

****

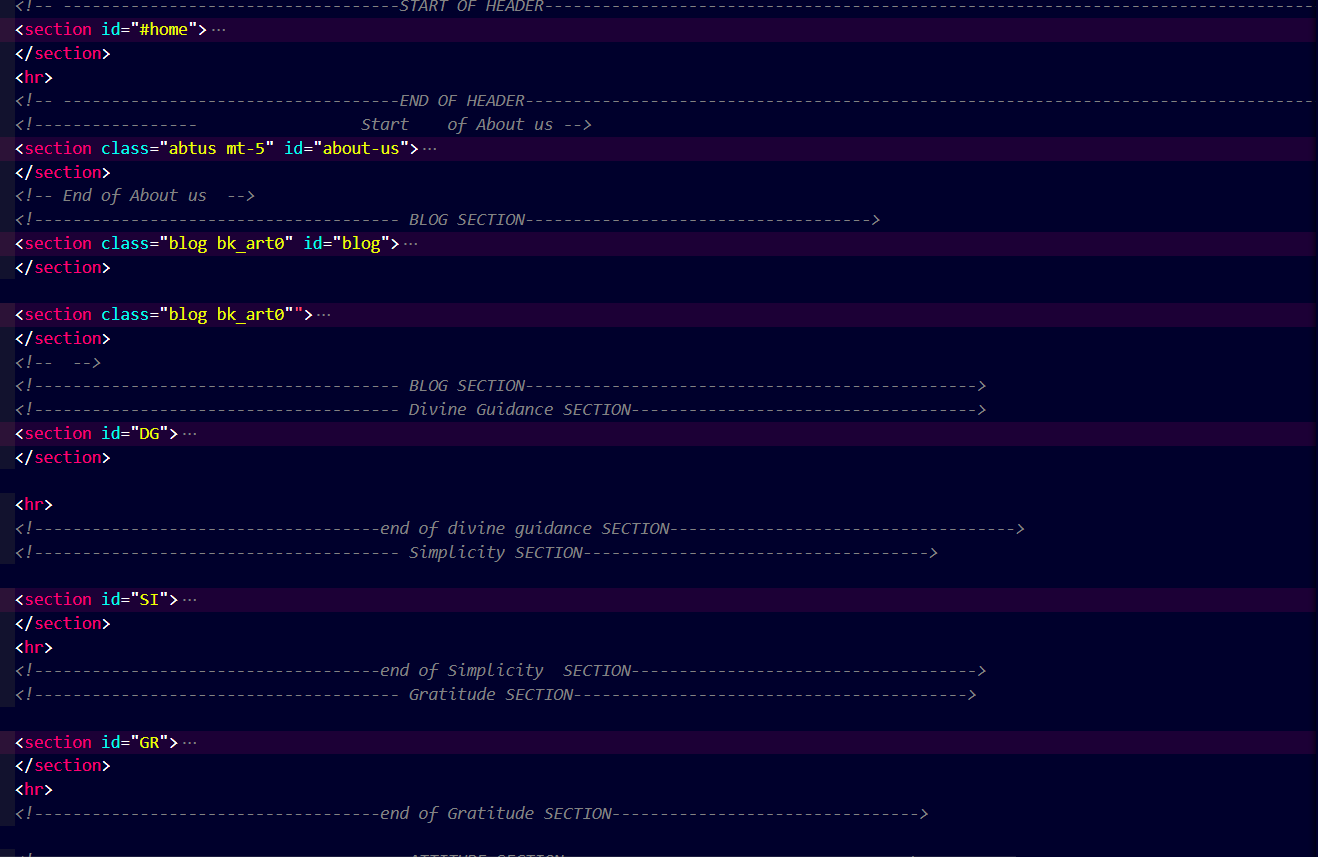
* The code starts with a <nav> tag and then has an opening <a class="logo" href="#home">Divine Door</a>.
* The code also has a closing </div> tag.
* The next part of the code is the main content area which starts with "HOME".
* It contains links to other pages in the website, such as BLOG, ABOUT US, and CONTACT.

**CSS**

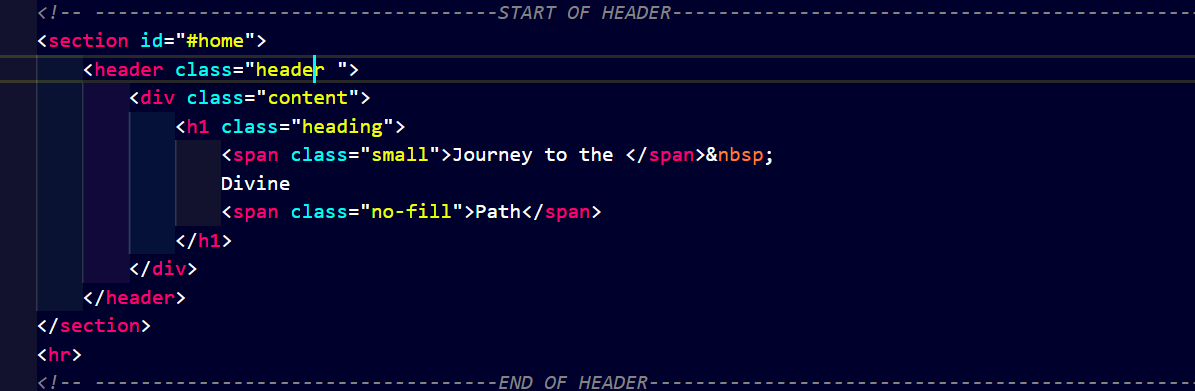
****

**Sections:**

1. Section – header
2. About us
3. Blog Card Section
4. Detailed Blog Card Description
5. Footer

****

## **1. Header**

****

* The code is a header with the text "Journey to the Divine Path" and an image of a path.
* The code is made up of two sections, one for the header and one for content.
* The header has a div that contains h1 tags with small font size, which are used as headings in this document.
* The code attempts to be a heading for the "Journey to the Divine Path" section.

**Output**

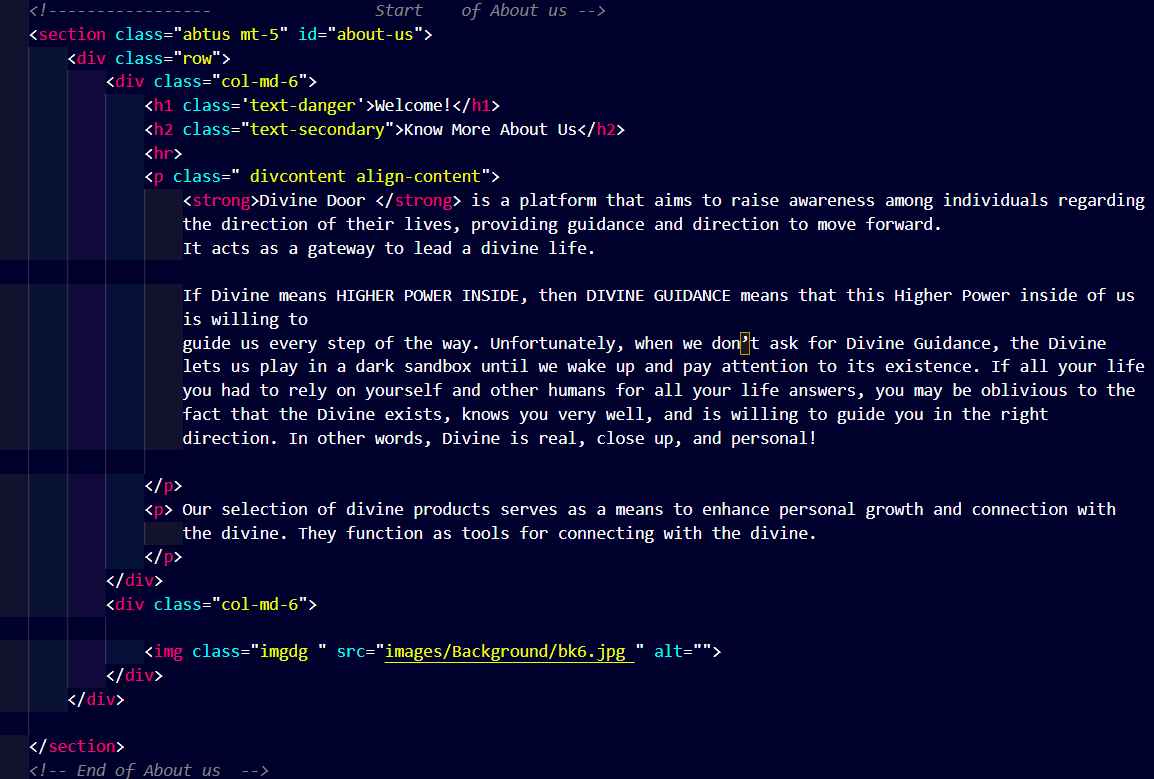
****

## **2. About us**

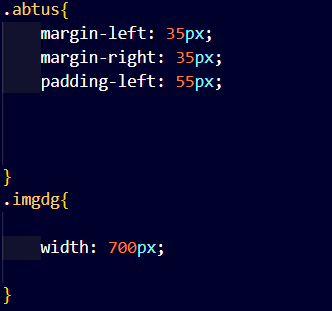
The information is about the company. Created this section using Tags :

<section>, <div> , <h1> , <hr> , <p> , <strong>

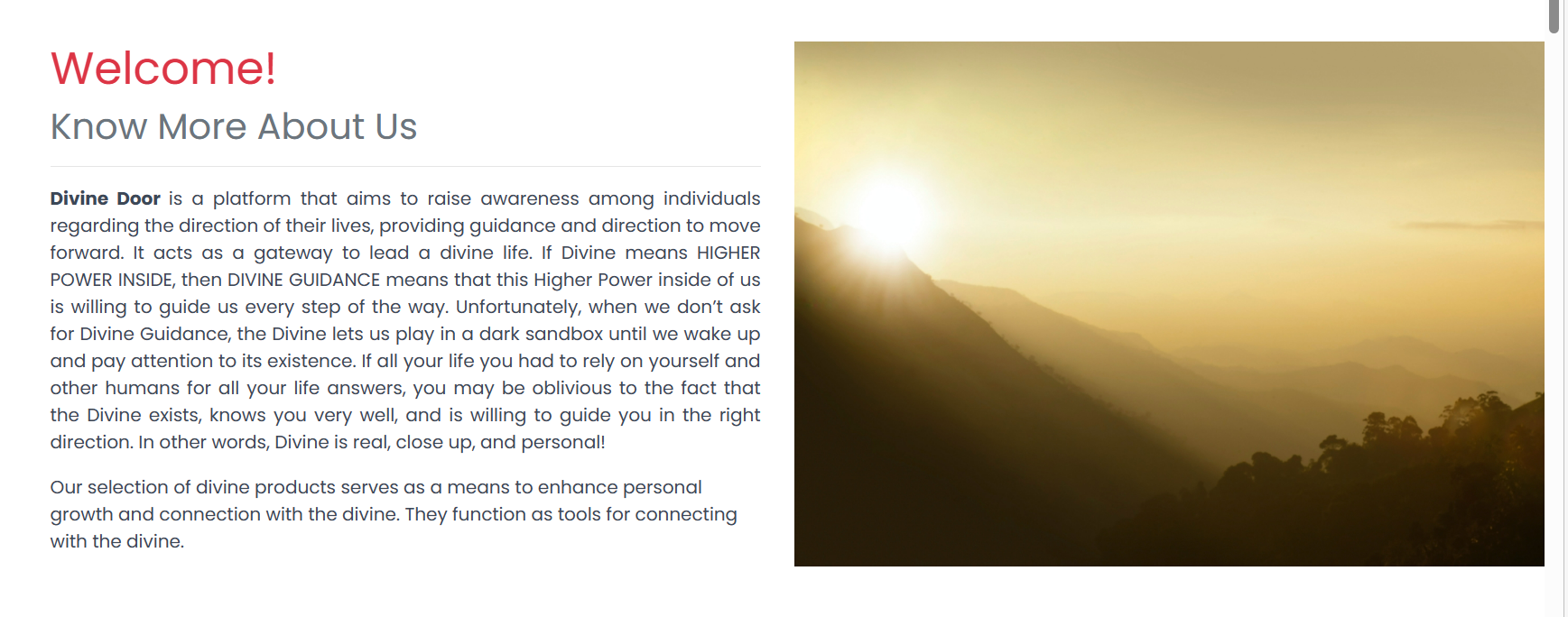
**HTML**

****

**CSS Properties**



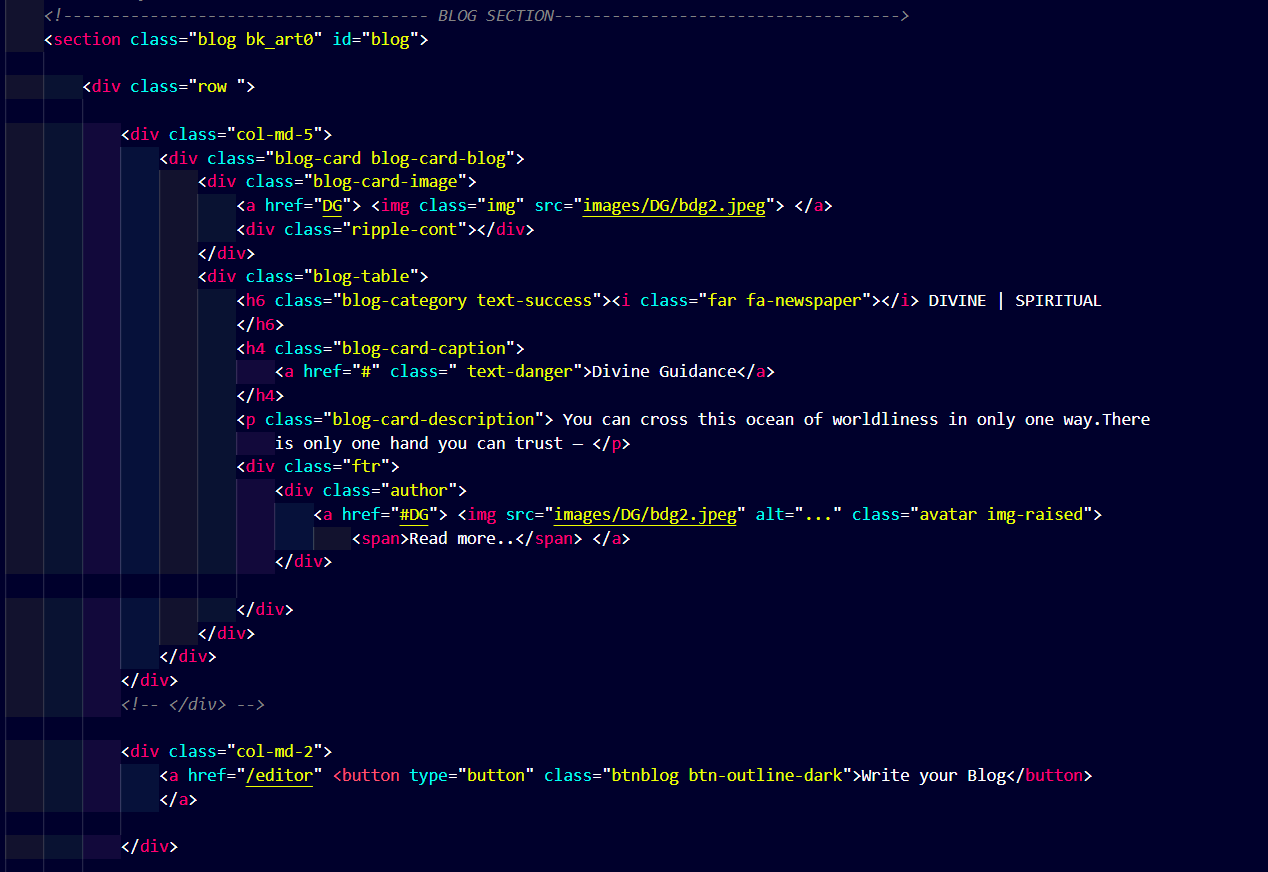
Output:

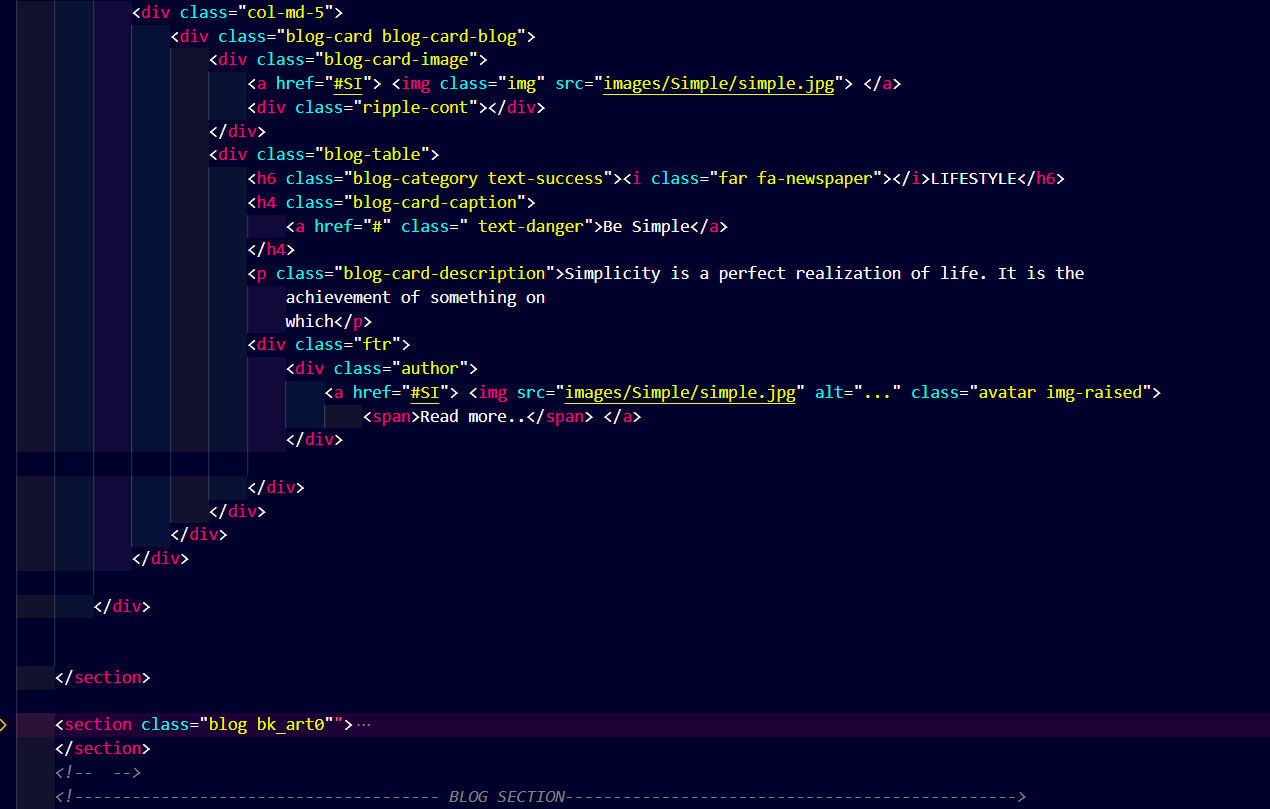


## **3. Blog Card Section**

It contains four published blog articles with images in a descriptive manner. Used tags bootstrap grid system, blog cards , images , tags(section , div, a ) with css properties

**HTML**

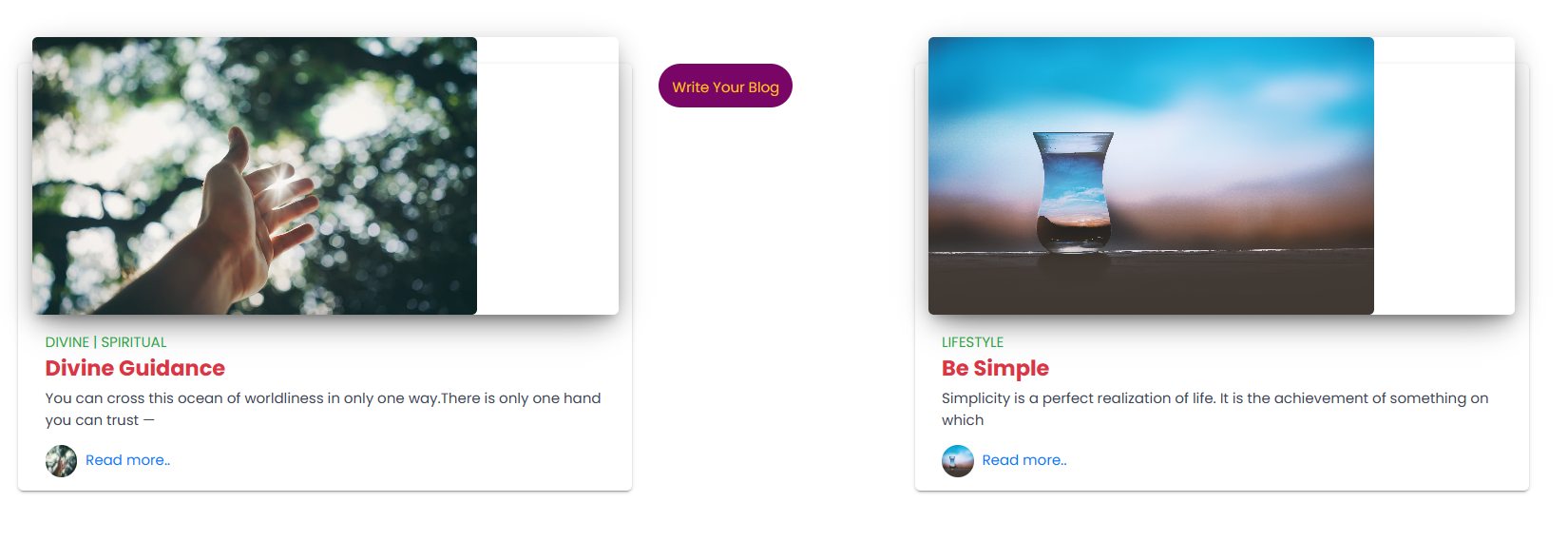
****

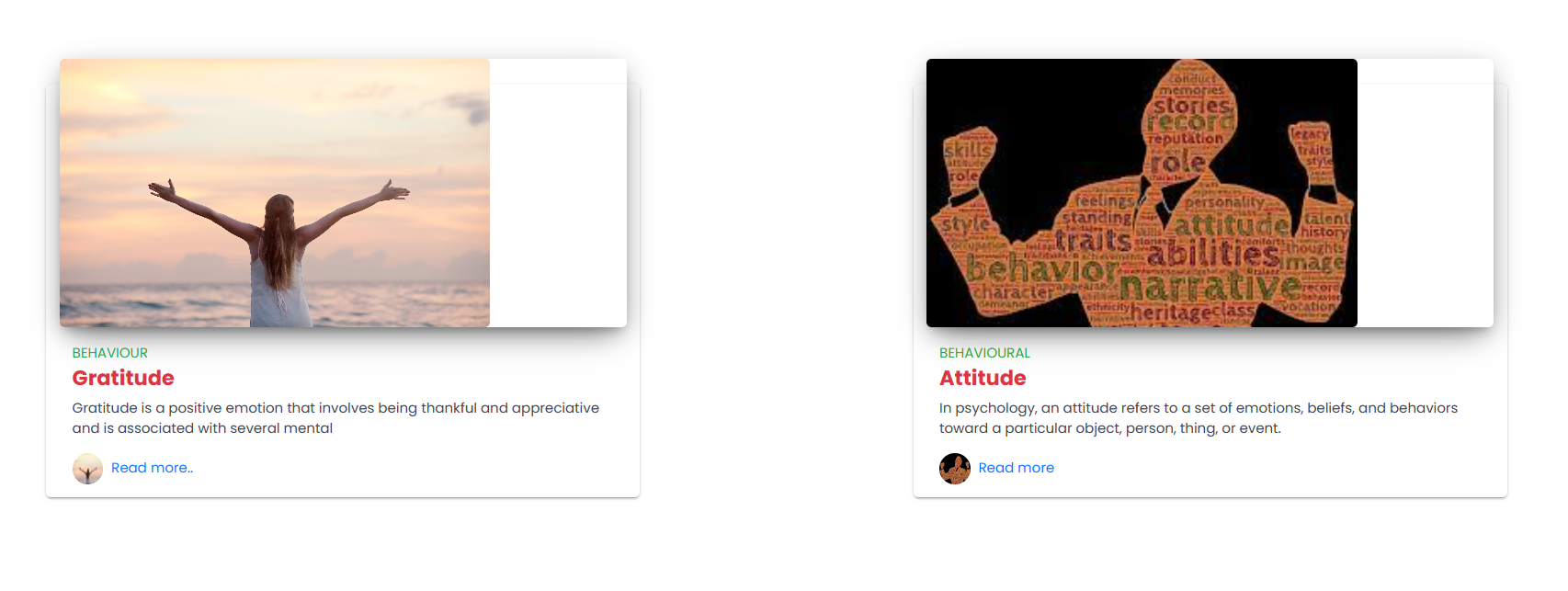
****

**CSS Properties**

****

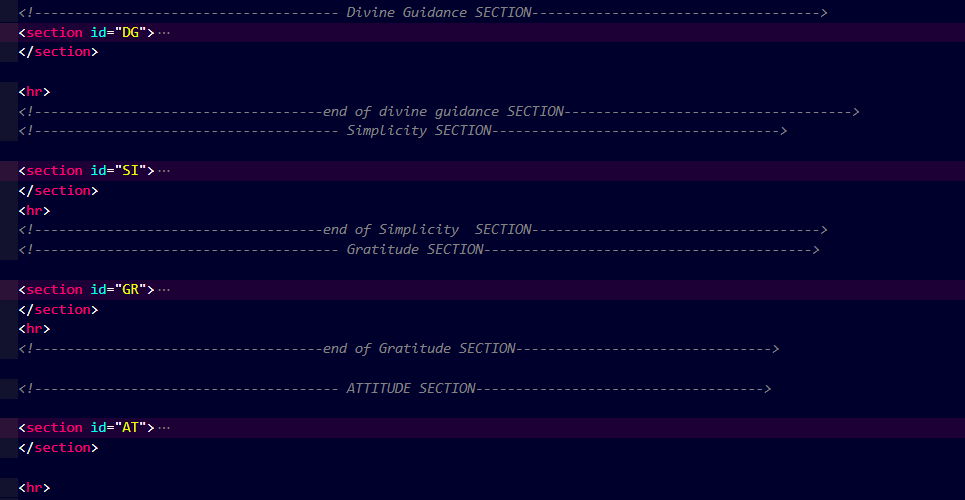
**Output**

****

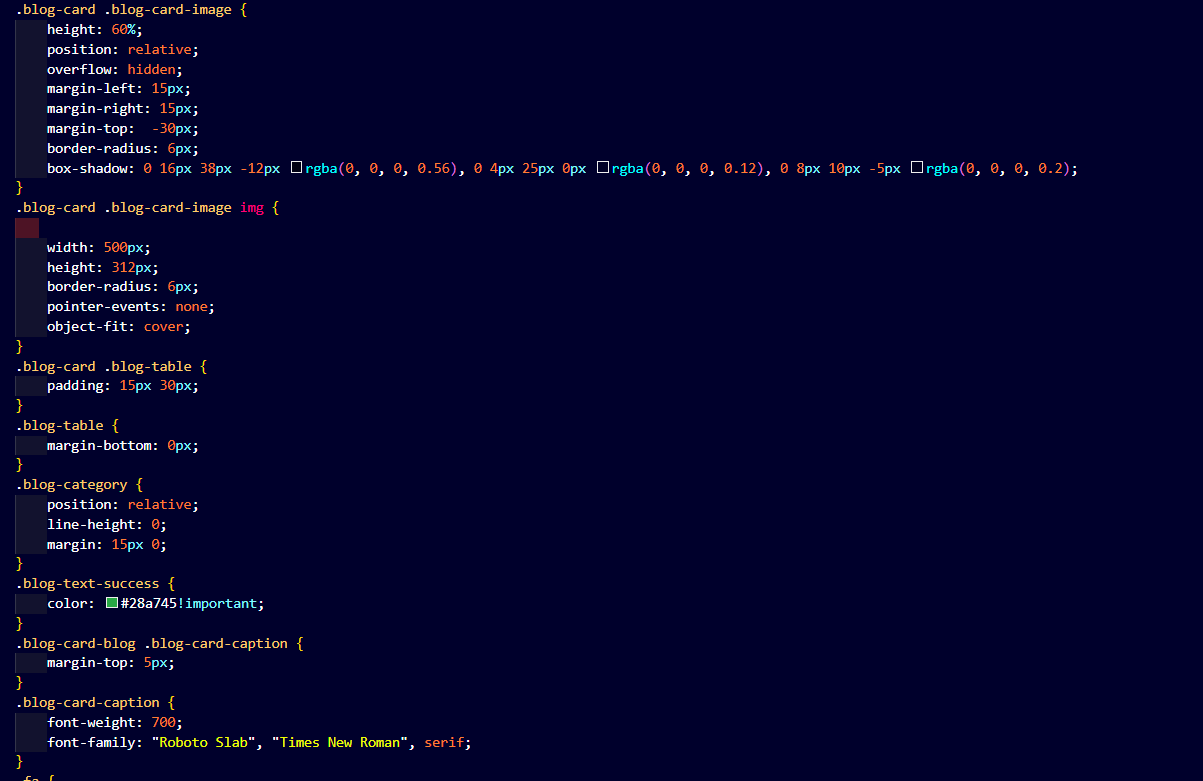
****

## **4. Detailed Blog Sections**

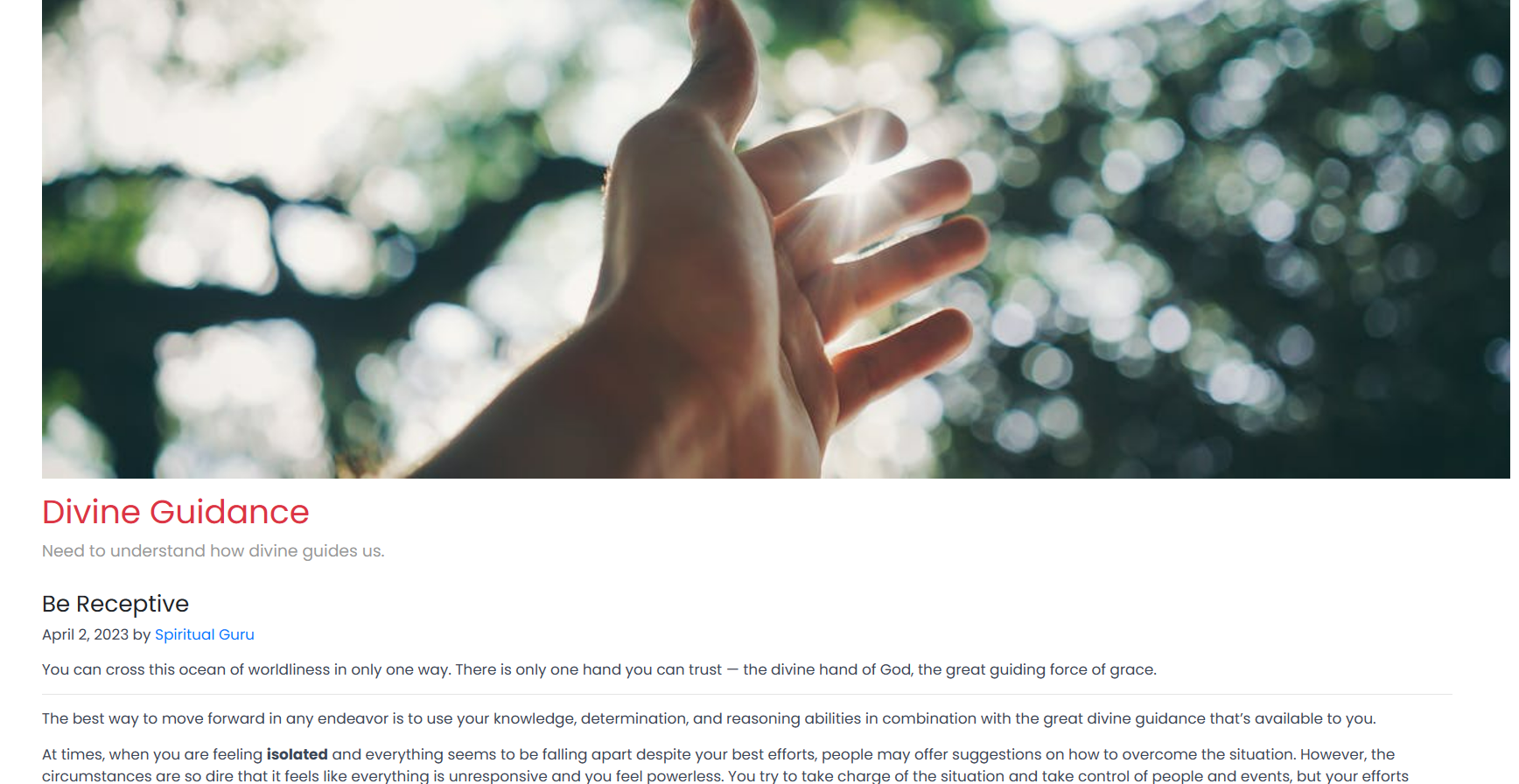
Created four sections with ids , images and description using tags p,section , div , hr with css properties.

HTML ****

CSS Properties

****

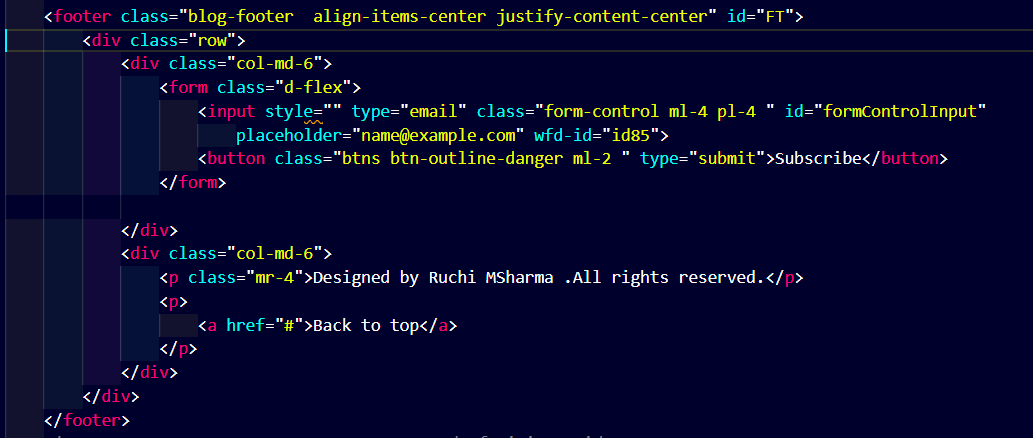
**Output**

****

## **5. Footer :**

The code starts with footer of blog post . It has two columns . first is for email input and the other contains information about designer and link for top of the page.

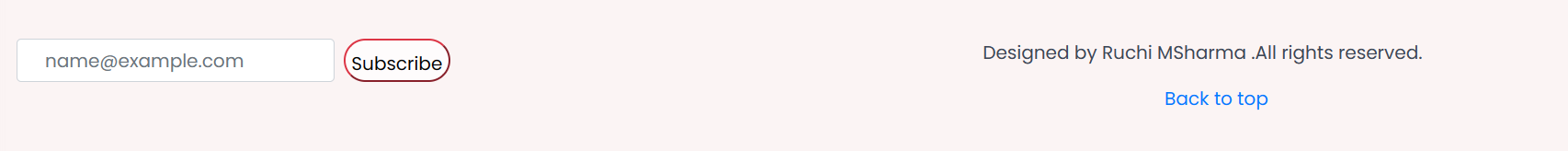
**HTML**

****

**CSS**

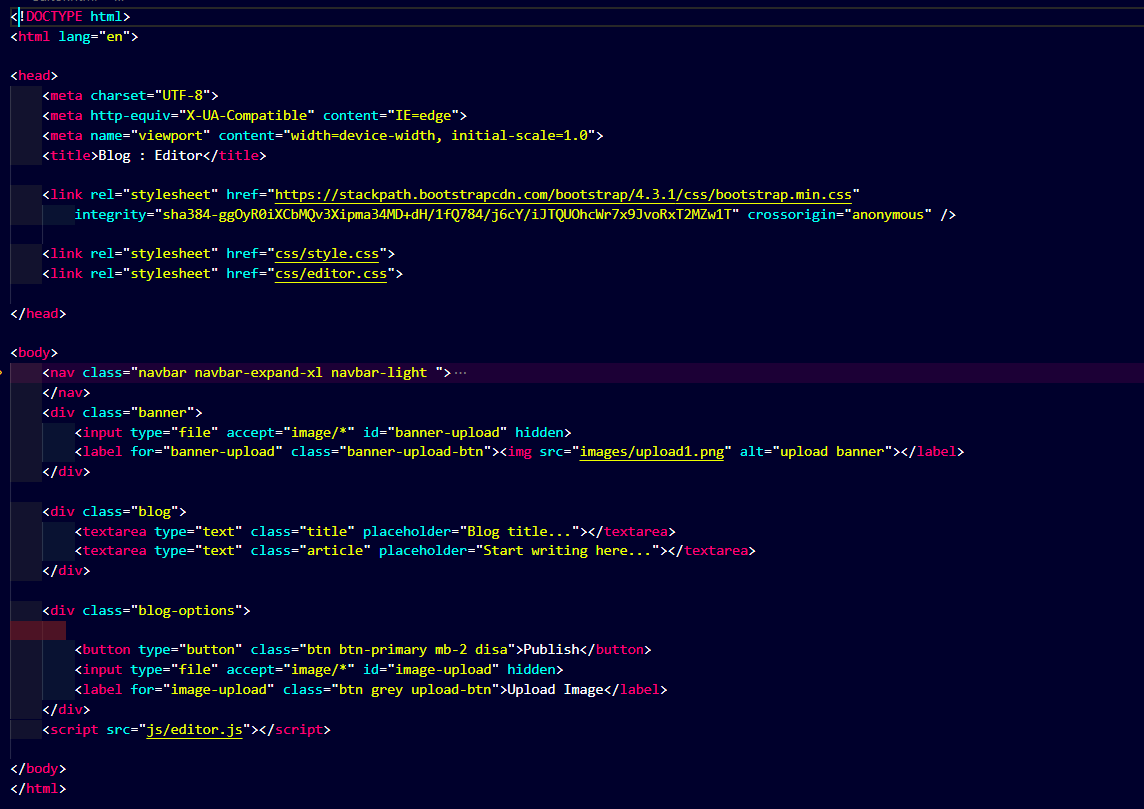
****

**Output**

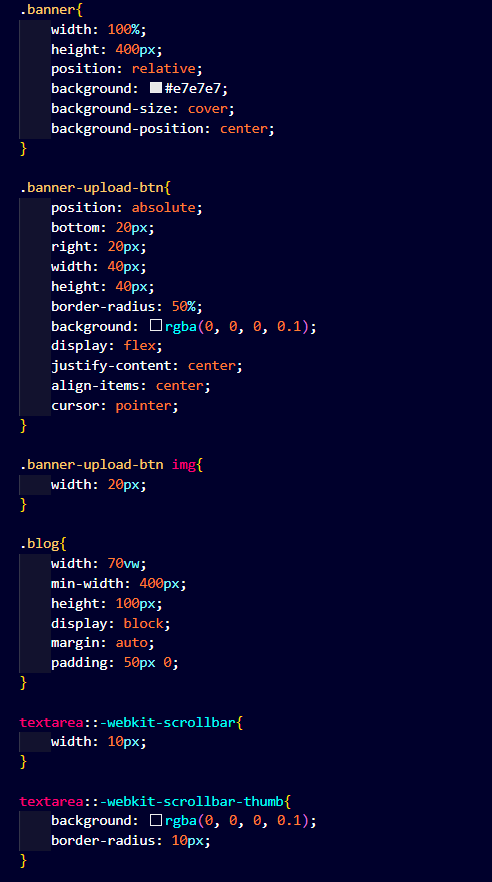
****

**Editor.html**

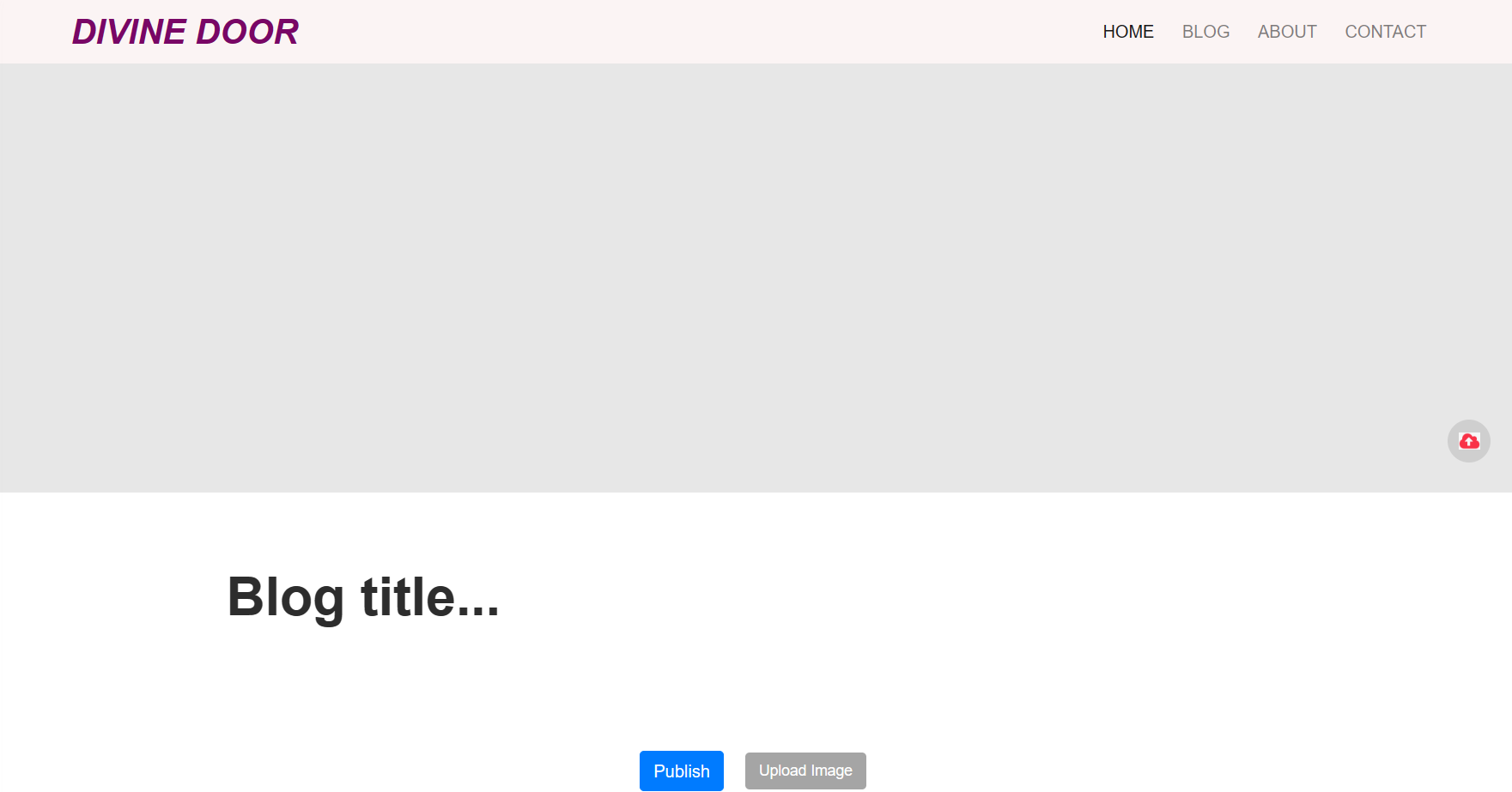
This page is shown on screen , when the user clicks button named “ Write Your blog” on home.html The main purpose of this file is to write a blog. It contains Nav bar along with it , option to upload the image file ,preview it , and store in the uploads folder in the project directory.



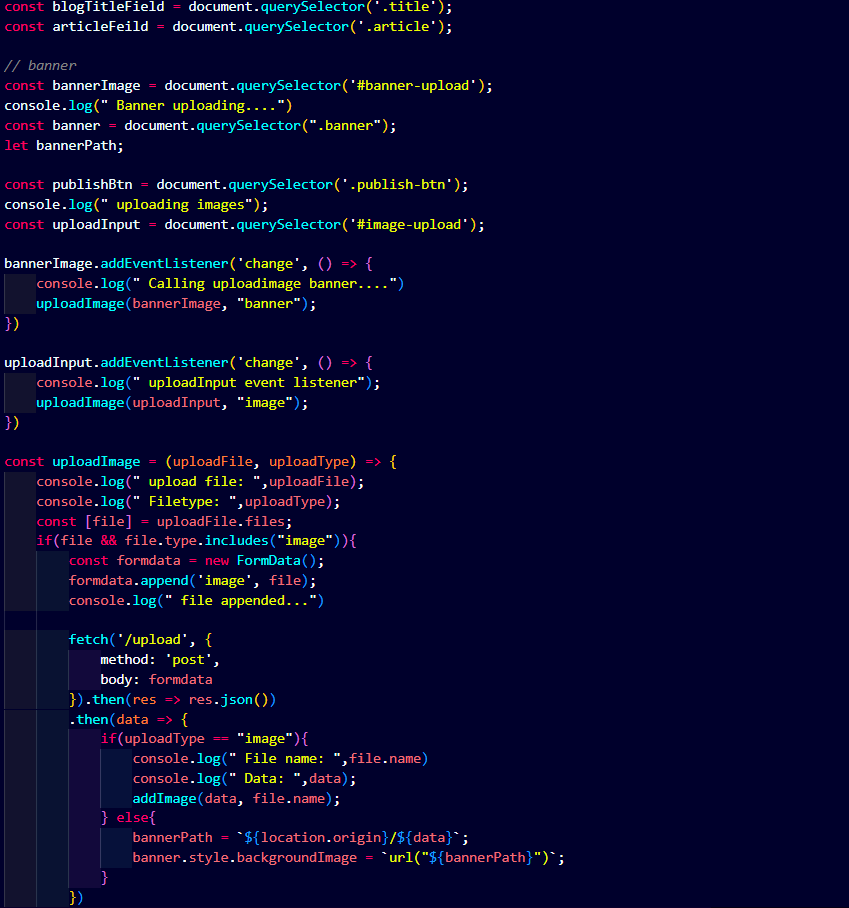
CSS:



**Output :**

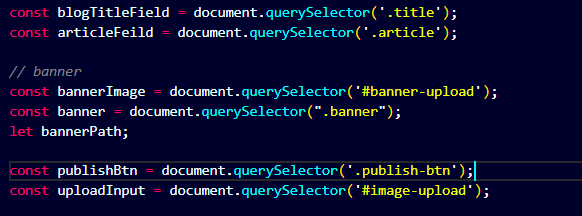
****

**Editor.js**

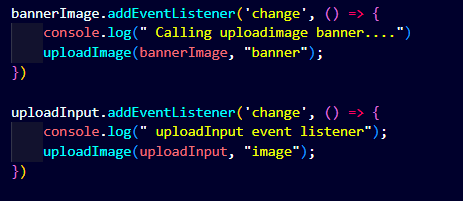
****

**Description**

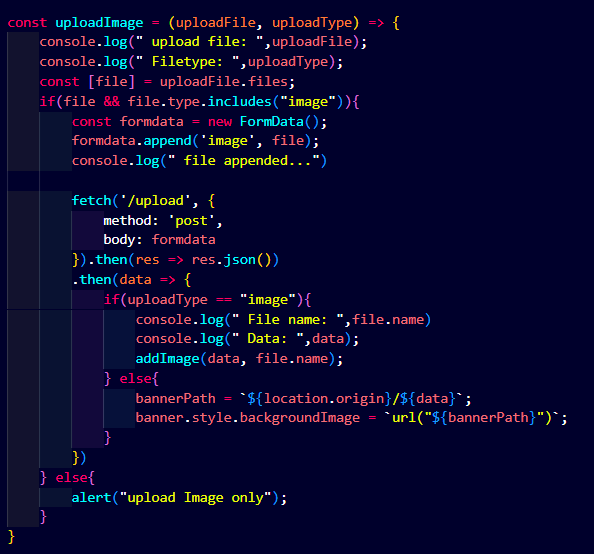
Start by selecting all elements that we need.

****

After selecting all elements. Add change event to our upload inputs and process the upload.



Now create uploadImage function.

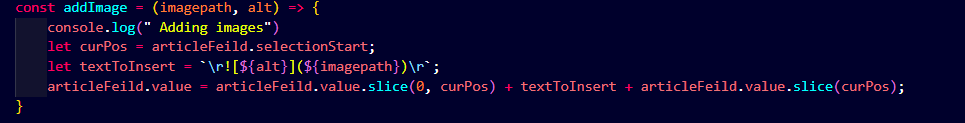


So this is how we can make our upload work. But it'll not work now because we haven't made our /upload route. For that open server.js and make /upload route.

Server.js

****

**Define a function in editor.js**

****

This function will let you insert a text format of your image for example if I upload 1.png then this function insert something like this ![1.png](image path) inside our article field.