Term Report

01286233 Web Programming

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

By

65011277 Chanasorn Howattanakulphong

65011320 Kanokjan Singhsuwan

65011381 Napatr Sapprasert

Table of contents

Source codes

Introduction

Our web project aimed to create an accessible and engaging online platform for the Software Engineering Department. Using HTML and CSS, we structured the website for easy navigation and a visually appealing experience.

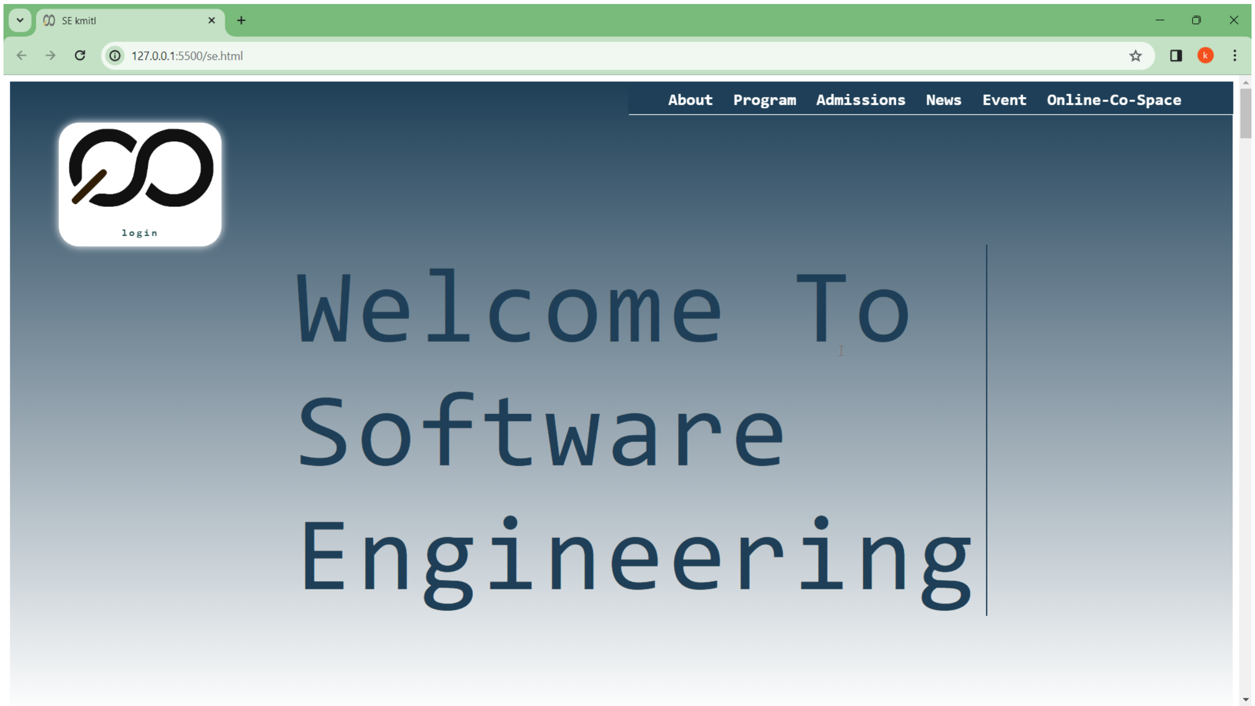
Upon user login, we implemented JavaScript to make the platform interactive. This allowed students to engage in real-time discussions, ask questions, and share insights on the blog section.

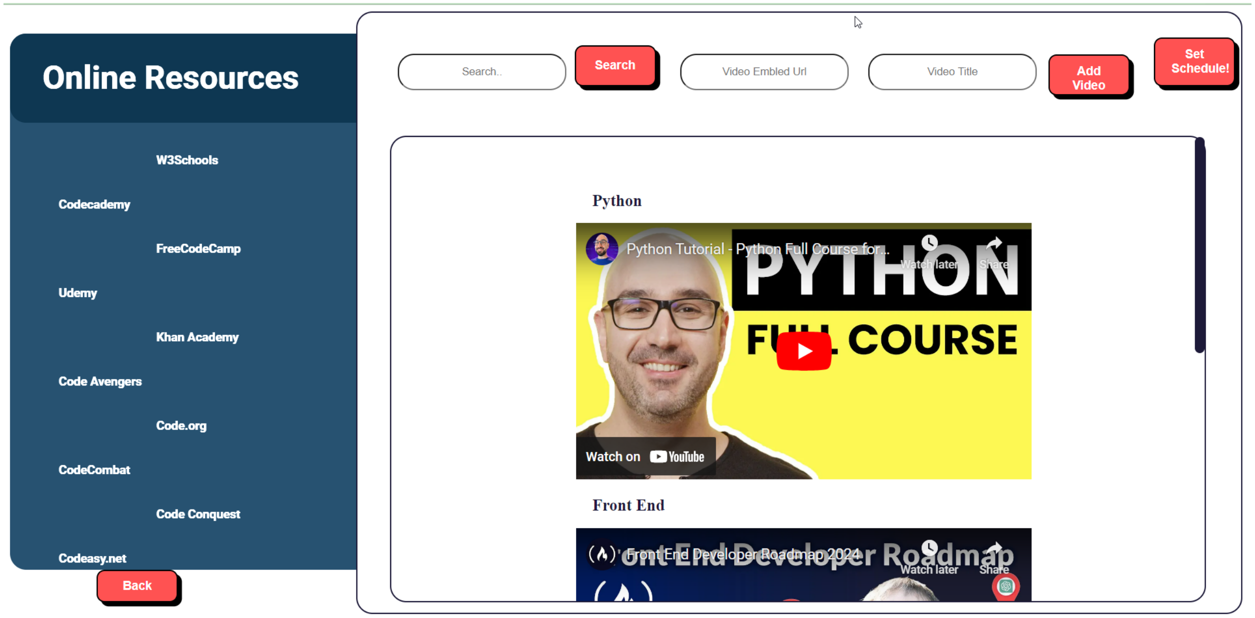
For the technical aspects, we utilized Uvicorn and the FastAPI Python library to manage the server efficiently. These tools ensured seamless operations, enabling users to access information without interruptions.

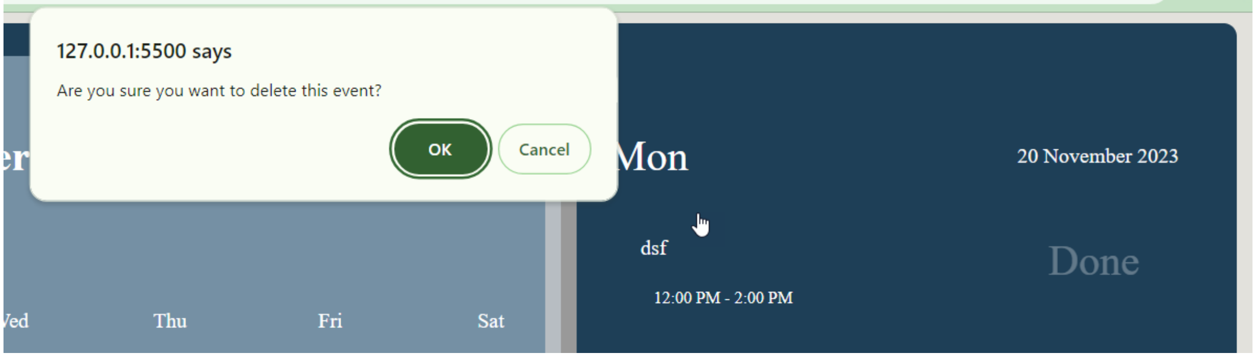
Additionally, we integrated a customizable calendar feature. This functionality enables users to personalize and manage important dates, facilitating organized planning and coordination within the department.

This report provides an in-depth overview of our development process, the challenges encountered, and the innovative solutions implemented. Our primary objective was to create an easily accessible platform that encourages active participation and facilitates effective organization within the Software Engineering Department.

Page Showcases







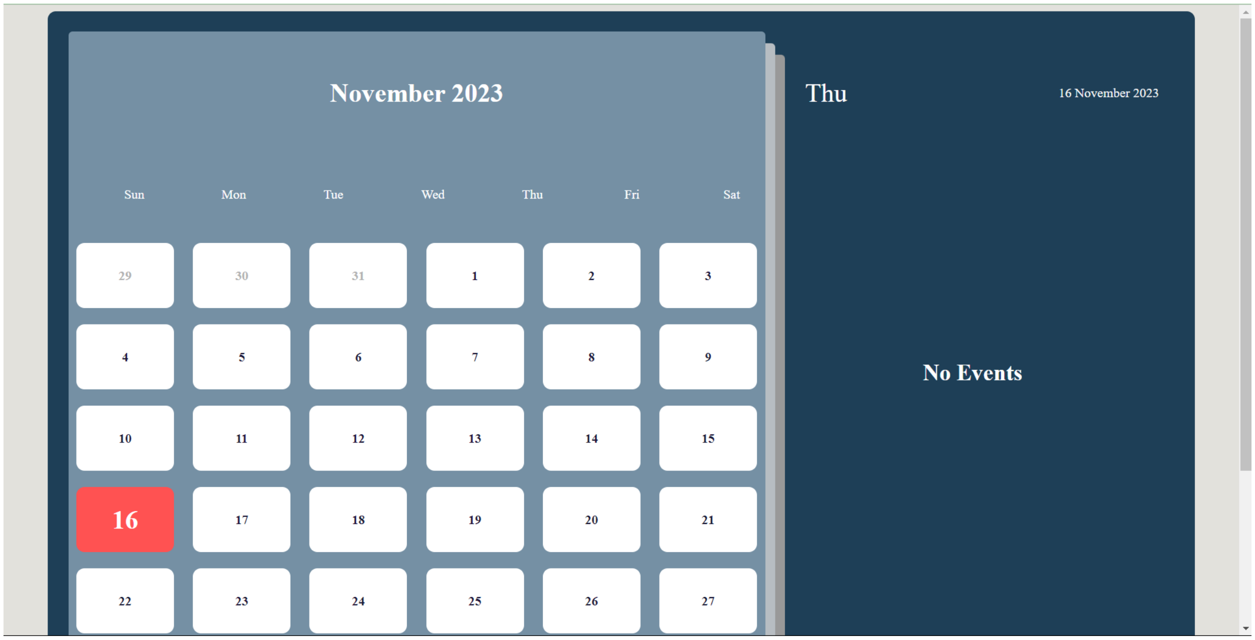
127.0.0.1:5500 says

Are you sure you want to delete this event?

Mon 20 November 2023

dsf

12:00 PM - 2:00 PM

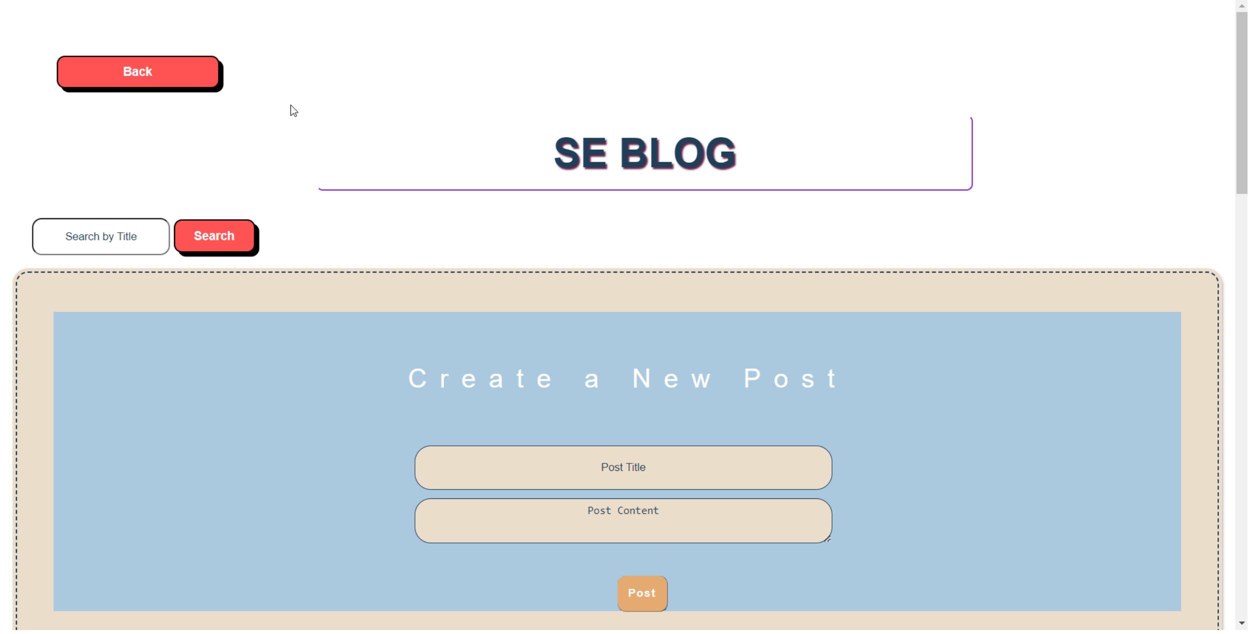


November 2023

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	
4	5	6	7	8	9	
10	11	12	13	14	15	
16	17	18	19	20	21	
22	23	24	25	26	27	

Thu 16 November 2023

No Events



SE BLOG

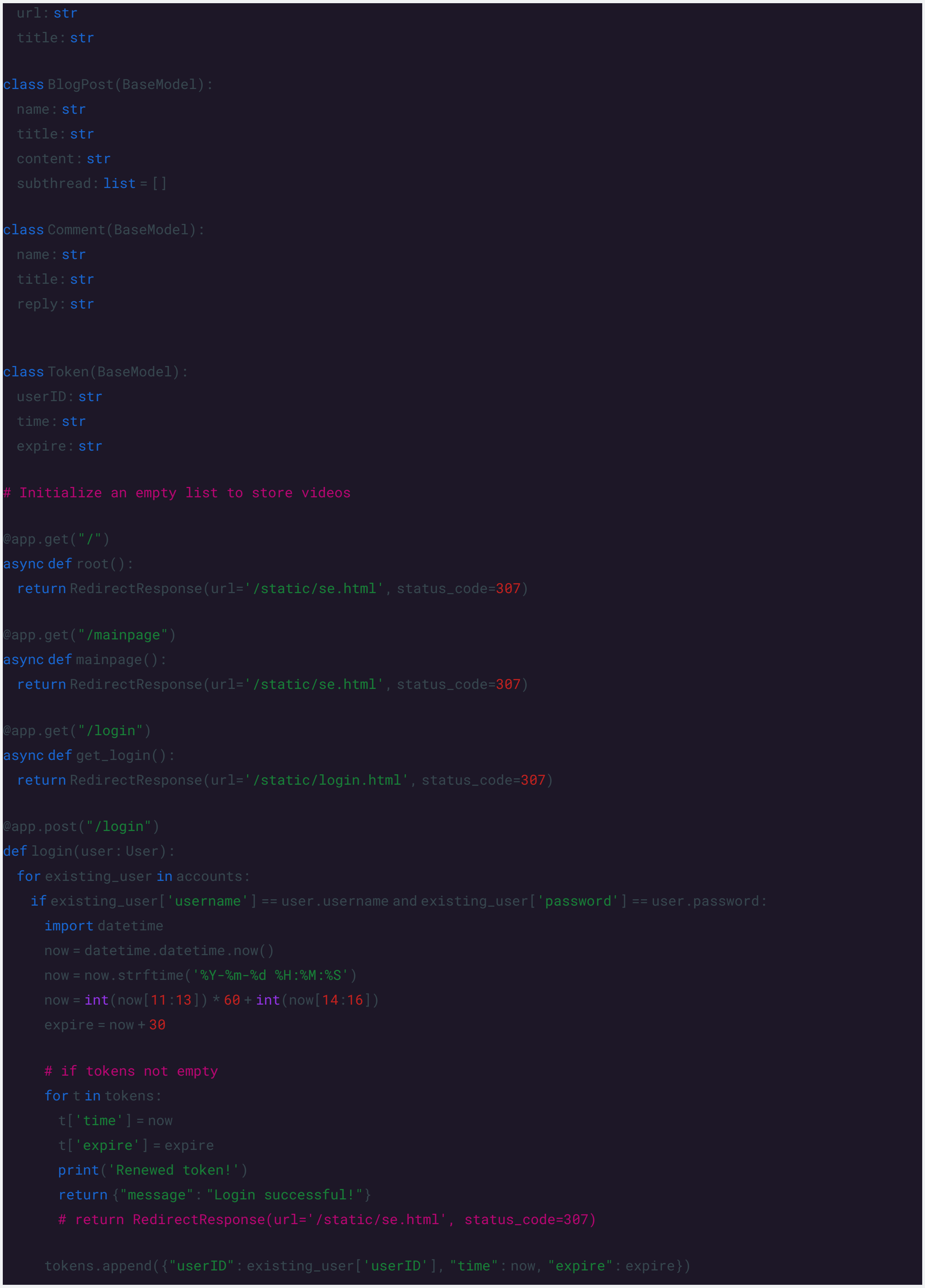
Create a New Post

Source Codes

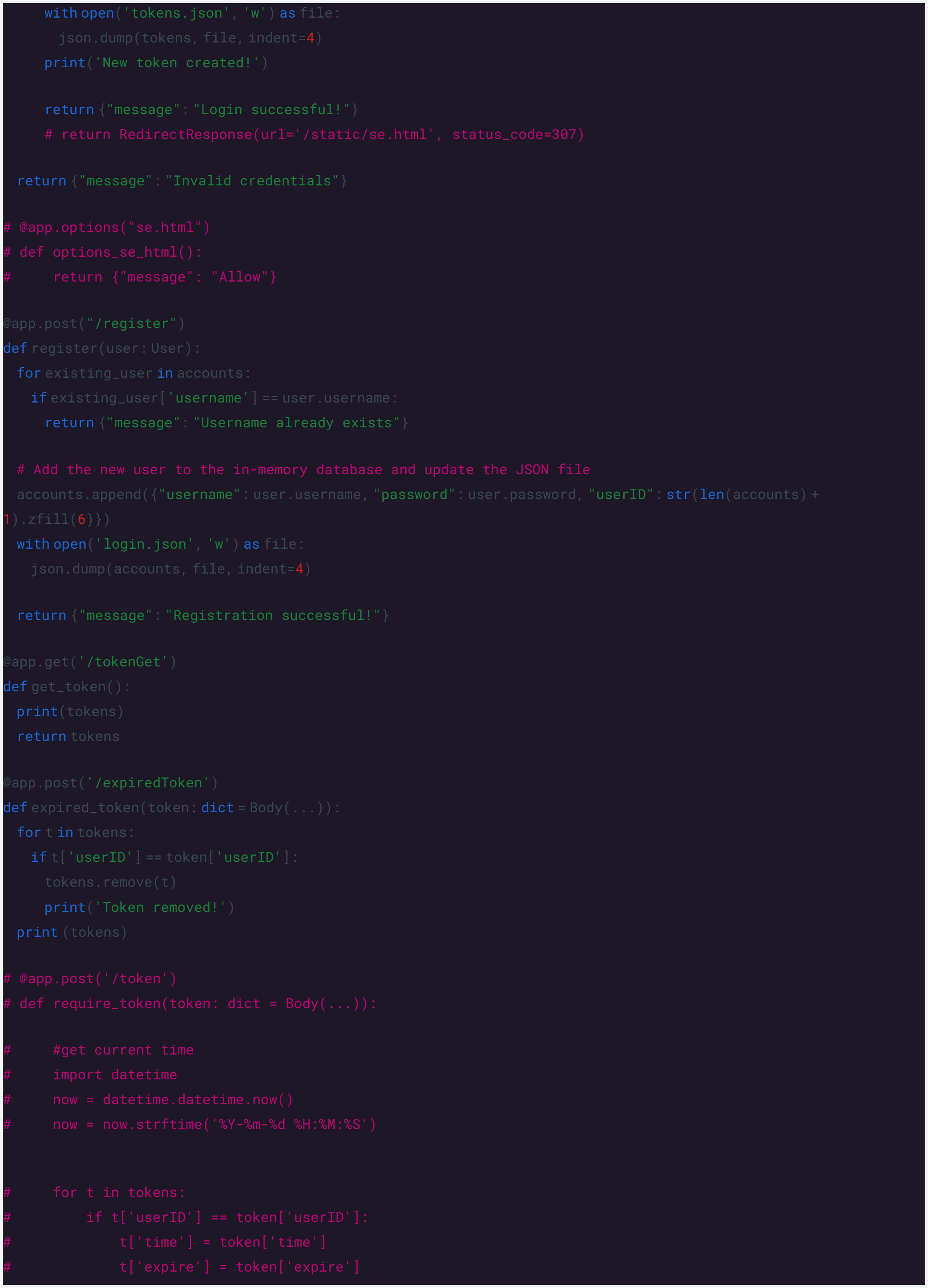
Server.py

Python

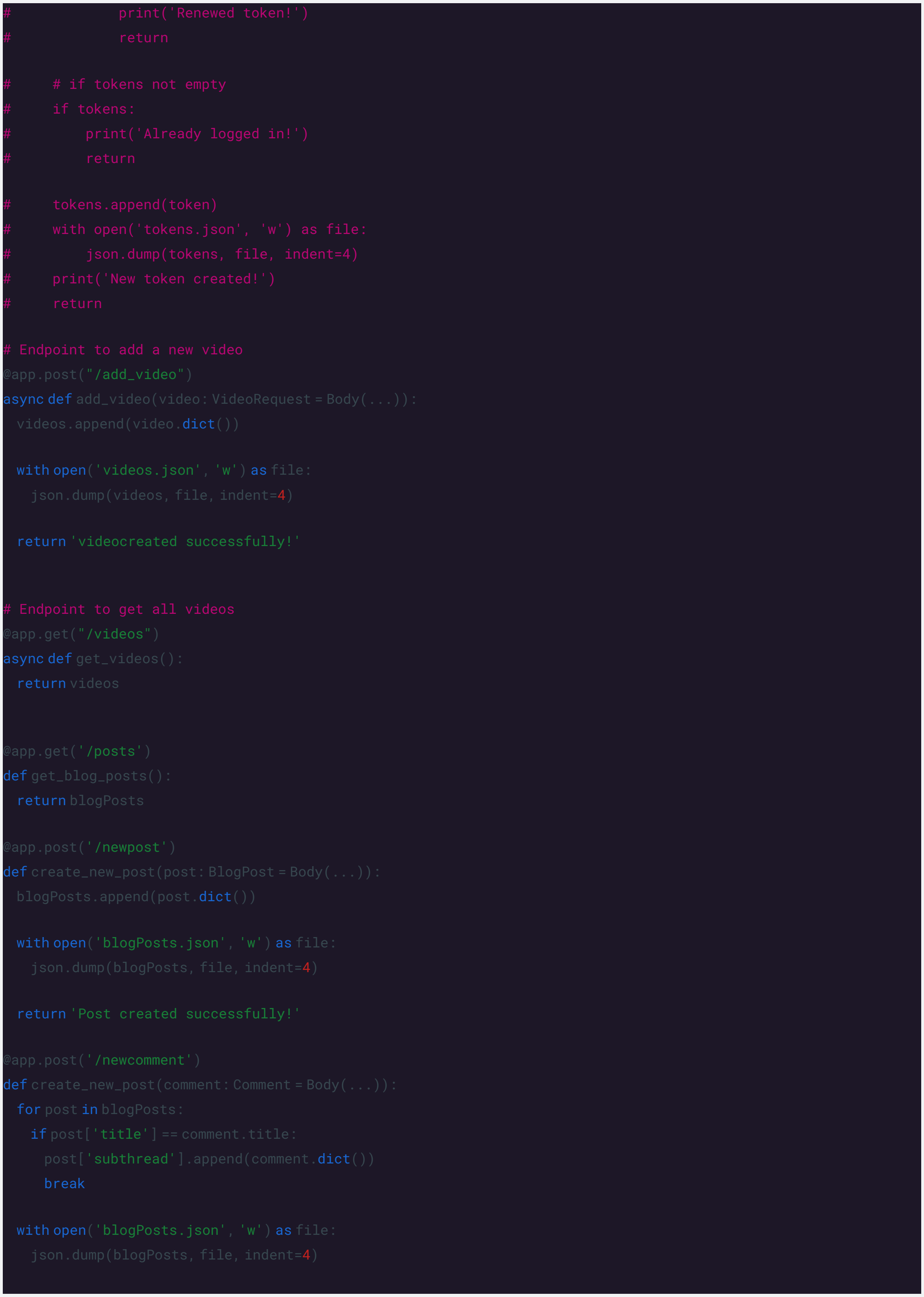
```python
from fastapi import FastAPI, HTTPException, Body
from pydantic import BaseModel
import json
from fastapi.middleware.cors import CORSMiddleware
from fastapi.staticfiles import StaticFiles
from fastapi.responses import RedirectResponse, HTMLResponse
# import os

app = FastAPI()
app.add_middleware(
  CORSMiddleware,
  allow_origins=["*"],
  allow_credentials=True,
  allow_methods=["*"],
  allow_headers=["*"],
)

# static_dir = os.path.join(os.path.dirname(os.path.realpath(__file__)), "static")

# Mount the static files directory to the "/static" path
app.mount("/static", StaticFiles(directory="static"), name="static")

# Mount another directory for resources to the "/resources" path
app.mount("/resources", StaticFiles(directory="resources"), name="resources")

# Load existing blog posts
with open('blogPosts.json', 'r') as file:
  blogPosts = json.load(file)

# Load existing accounts
with open('login.json', 'r') as file:
  accounts = json.load(file)

# Load existing tokens
with open('tokens.json', 'r') as file:
  tokens = json.load(file)


with open('videos.json', 'r') as file:
  videos = json.load(file)

class User(BaseModel):
  username: str
  password: str

class VideoRequest(BaseModel):
```



```
  url: str
  title: str

class BlogPost(BaseModel):
  name: str
  title: str
  content: str
  subthread: list = []

class Comment(BaseModel):
  name: str
  title: str
  reply: str


class Token(BaseModel):
  userID: str
  time: str
  expire: str

# Initialize an empty list to store videos

@app.get("/")
async def root():
  return RedirectResponse(url='/static/se.html', status_code=307)

@app.get("/mainpage")
async def mainpage():
  return RedirectResponse(url='/static/se.html', status_code=307)

@app.get("/login")
async def get_login():
  return RedirectResponse(url='/static/login.html', status_code=307)

@app.post("/login")
def login(user: User):
  for existing_user in accounts:
    if existing_user['username'] == user.username and existing_user['password'] == user.password:
      import datetime
      now = datetime.datetime.now()
      now = now.strftime('%Y-%m-%d %H:%M:%S')
      now = int(now[11:13]) * 60 + int(now[14:16])
      expire = now + 30

      # if tokens not empty
      for t in tokens:
        t['time'] = now
        t['expire'] = expire
        print('Renewed token!')
        return {"message": "Login successful!"}
        # return RedirectResponse(url='/static/se.html', status_code=307)

      tokens.append({"userID": existing_user['userID'], "time": now, "expire": expire})
```



```
      with open('tokens.json', 'w') as file:
        json.dump(tokens, file, indent=4)
      print('New token created!')

      return {"message": "Login successful!"}
      # return RedirectResponse(url='/static/se.html', status_code=307)

  return {"message": "Invalid credentials"}

# @app.options("se.html")
# def options_se_html():
#     return {"message": "Allow"}

@app.post("/register")
def register(user: User):
  for existing_user in accounts:
    if existing_user['username'] == user.username:
      return {"message": "Username already exists"}

  # Add the new user to the in-memory database and update the JSON file
  accounts.append({"username": user.username, "password": user.password, "userID": str(len(accounts) +
1).zfill(6)})
  with open('login.json', 'w') as file:
    json.dump(accounts, file, indent=4)

  return {"message": "Registration successful!"}

@app.get('/tokenGet')
def get_token():
  print(tokens)
  return tokens

@app.post('/expiredToken')
def expired_token(token: dict = Body(...)):
  for t in tokens:
    if t['userID'] == token['userID']:
      tokens.remove(t)
      print('Token removed!')
  print (tokens)

# @app.post('/token')
# def require_token(token: dict = Body(...)):

#     #get current time
#     import datetime
#     now = datetime.datetime.now()
#     now = now.strftime('%Y-%m-%d %H:%M:%S')


#     for t in tokens:
#         if t['userID'] == token['userID']:
#             t['time'] = token['time']
#             t['expire'] = token['expire']
```

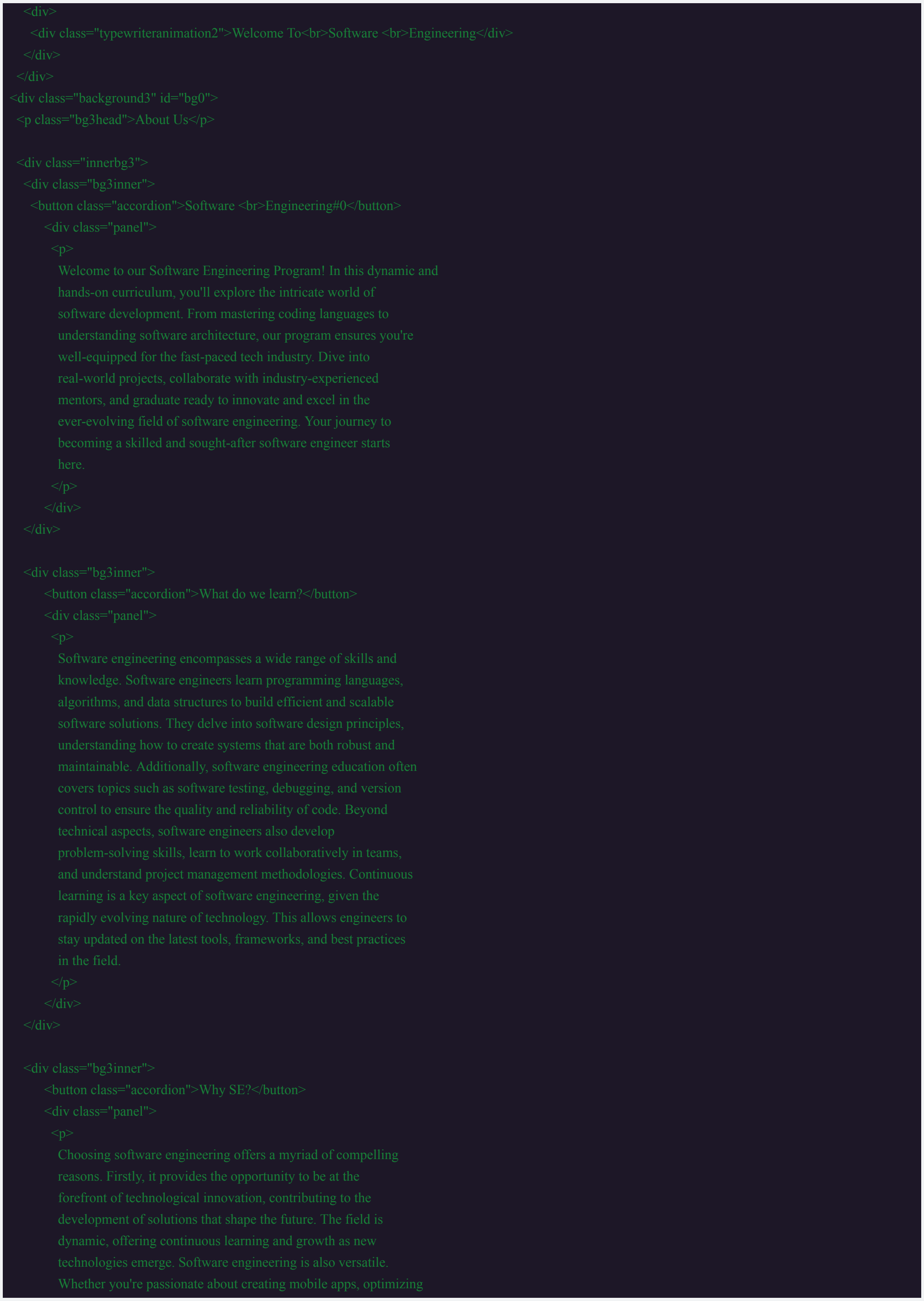


```
#             print('Renewed token!')
#             return

#     # if tokens not empty
#     if tokens:
#         print('Already logged in!')
#         return

#     tokens.append(token)
#     with open('tokens.json', 'w') as file:
#         json.dump(tokens, file, indent=4)
#     print('New token created!')
#     return

# Endpoint to add a new video
@app.post("/add_video")
async def add_video(video: VideoRequest = Body(...)):
  videos.append(video.dict())

  with open('videos.json', 'w') as file:
    json.dump(videos, file, indent=4)

  return 'videocreated successfully!'


# Endpoint to get all videos
@app.get("/videos")
async def get_videos():
  return videos


@app.get('/posts')
def get_blog_posts():
  return blogPosts

@app.post('/newpost')
def create_new_post(post: BlogPost = Body(...)):
  blogPosts.append(post.dict())

  with open('blogPosts.json', 'w') as file:
    json.dump(blogPosts, file, indent=4)

  return 'Post created successfully!'

@app.post('/newcomment')
def create_new_post(comment: Comment = Body(...)):
  for post in blogPosts:
    if post['title'] == comment.title:
      post['subthread'].append(comment.dict())
      break

  with open('blogPosts.json', 'w') as file:
    json.dump(blogPosts, file, indent=4)
  return 'Post created successfully!'

@app.get('/search/')
def search_posts(query: str):
  filtered_posts = [post for post in blogPosts if query.lower() in post['title'].lower()]
  return filtered_posts

@app.get('/searchvid/')
def search_video(query: str):
  filtered_video = [video for video in videos if query.lower() in video['title'].lower()]
  return filtered_video

if __name__ == '__main__':
  import uvicorn
  uvicorn.run(app, host='127.0.0.1', port=8000)
```

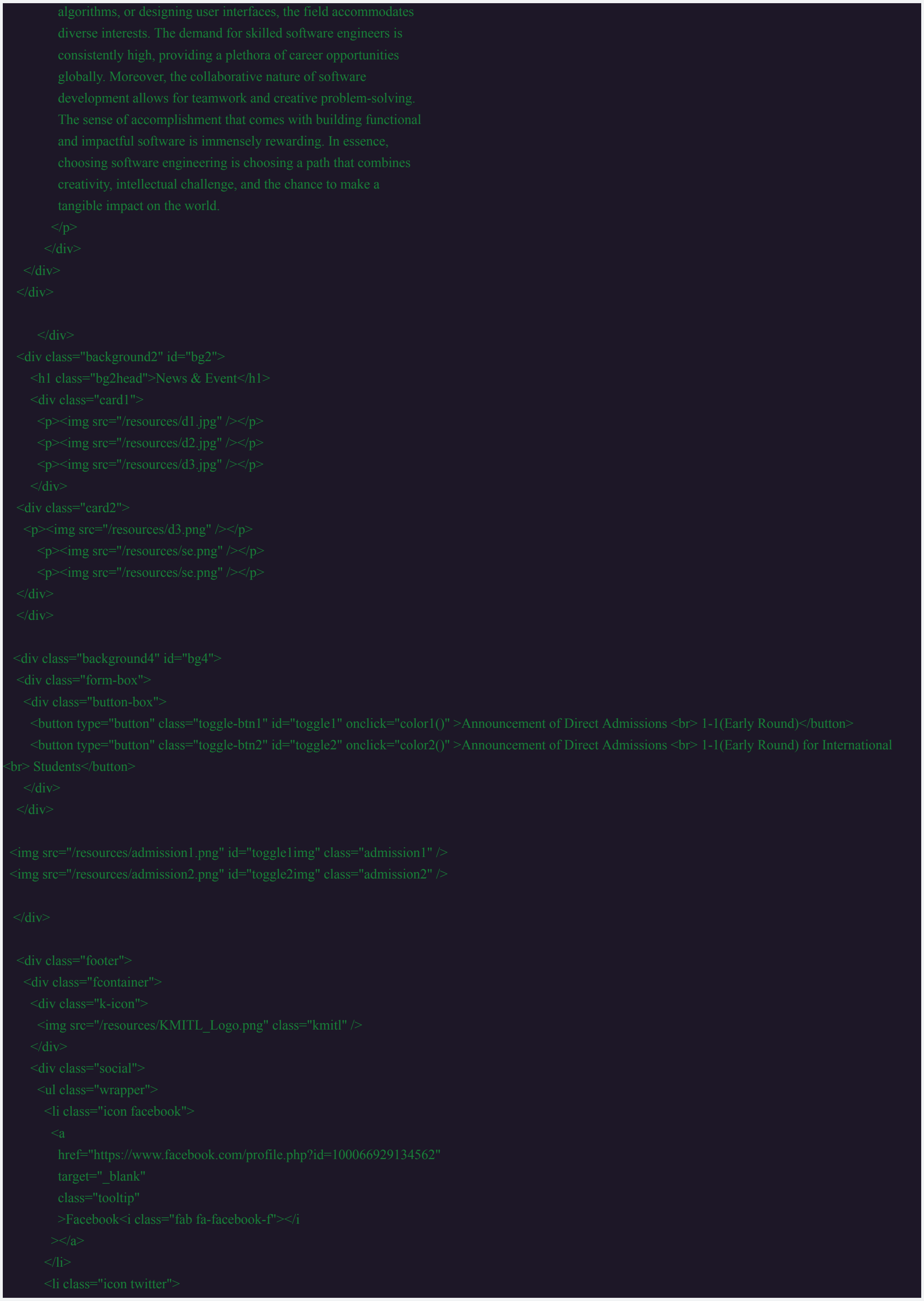
Se.html

Unset

```
<!DOCTYPE html>
<html lang="en">
 <style type="text/css"></style>
 <head>
  <meta charset="UTF-8" />
  <link rel="icon" type="image/x-icon" href="/resources/se.png" />

  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <link rel="stylesheet" href="se.css" />
  <script src="/static/se.js"></script>

  <title>SE kmitl</title>
 </head>
 <body>
  <script src="/static/se.js"></script>

  <div class="loader">
   <div class="loader1">
    <div class="dot"></div>
    <div class="dot"></div>
    <div class="dot"></div>
   </div>
  </div>
  <div class="background1" id="bg1" ">
   <div class="item1">
    <div id="login-div">
     <img src="/resources/se.png" alt="se" class="se" />

     <p id="login-text">login</p>
    </div>
    <br />
   </div>
   <div class="tag">
    <div>
     <a class="navitem" href="#bg0">About</a>
    </div>
    <div>
     <a class="navitem" href="#bg1">Program</a>
    </div>
    <div>
     <a class="navitem" href="#bg4">Admissions</a>
    </div>
    <div>
     <a class="navitem" href="#bg2">News</a>
    </div>
    <div>
     <a class="navitem" href="#bg2">Event</a>
    </div>

    <div class="dropdown">
     Online-Co-Space
     <div class="dropdown-content">
      <a class="HHH" href="./online.html">Online Resource</a>
      <a class="HHH" href="./blog.html">Blog</a>
     </div>
    </div>
   </div>
```



```
  <div>
   <div class="typewriteranimation2">Welcome To<br>Software <br>Engineering</div>
  </div>
 </div>
<div class="background3" id="bg0">
 <p class="bg3head">About Us</p>

 <div class="innerbg3">
  <div class="bg3inner">
   <button class="accordion">Software <br>Engineering#0</button>
     <div class="panel">
      <p>
       Welcome to our Software Engineering Program! In this dynamic and
       hands-on curriculum, you'll explore the intricate world of
       software development. From mastering coding languages to
       understanding software architecture, our program ensures you're
       well-equipped for the fast-paced tech industry. Dive into
       real-world projects, collaborate with industry-experienced
       mentors, and graduate ready to innovate and excel in the
       ever-evolving field of software engineering. Your journey to
       becoming a skilled and sought-after software engineer starts
       here.
      </p>
     </div>
  </div>

  <div class="bg3inner">
     <button class="accordion">What do we learn?</button>
     <div class="panel">
      <p>
       Software engineering encompasses a wide range of skills and
       knowledge. Software engineers learn programming languages,
       algorithms, and data structures to build efficient and scalable
       software solutions. They delve into software design principles,
       understanding how to create systems that are both robust and
       maintainable. Additionally, software engineering education often
       covers topics such as software testing, debugging, and version
       control to ensure the quality and reliability of code. Beyond
       technical aspects, software engineers also develop
       problem-solving skills, learn to work collaboratively in teams,
       and understand project management methodologies. Continuous
       learning is a key aspect of software engineering, given the
       rapidly evolving nature of technology. This allows engineers to
       stay updated on the latest tools, frameworks, and best practices
       in the field.
      </p>
     </div>
  </div>

  <div class="bg3inner">
     <button class="accordion">Why SE?</button>
     <div class="panel">
      <p>
       Choosing software engineering offers a myriad of compelling
       reasons. Firstly, it provides the opportunity to be at the
       forefront of technological innovation, contributing to the
       development of solutions that shape the future. The field is
       dynamic, offering continuous learning and growth as new
       technologies emerge. Software engineering is also versatile.
       Whether you're passionate about creating mobile apps, optimizing
```



```
        algorithms, or designing user interfaces, the field accommodates
        diverse interests. The demand for skilled software engineers is
        consistently high, providing a plethora of career opportunities
        globally. Moreover, the collaborative nature of software
        development allows for teamwork and creative problem-solving.
        The sense of accomplishment that comes with building functional
        and impactful software is immensely rewarding. In essence,
        choosing software engineering is choosing a path that combines
        creativity, intellectual challenge, and the chance to make a
        tangible impact on the world.
       </p>
      </div>
   </div>
  </div>

     </div>
  <div class="background2" id="bg2">
    <h1 class="bg2head">News & Event</h1>
    <div class="card1">
     <p><img src="/resources/d1.jpg" /></p>
     <p><img src="/resources/d2.jpg" /></p>
     <p><img src="/resources/d3.jpg" /></p>
    </div>
  <div class="card2">
   <p><img src="/resources/d3.png" /></p>
     <p><img src="/resources/se.png" /></p>
     <p><img src="/resources/se.png" /></p>
  </div>
  </div>

  <div class="background4" id="bg4">
  <div class="form-box">
   <div class="button-box">
    <button type="button" class="toggle-btn1" id="toggle1" onclick="color1()" >Announcement of Direct Admissions <br> 1-1(Early Round)</button>
    <button type="button" class="toggle-btn2" id="toggle2" onclick="color2()" >Announcement of Direct Admissions <br> 1-1(Early Round) for International
<br> Students</button>
   </div>
  </div>

 <img src="/resources/admission1.png" id="toggle1img" class="admission1" />
 <img src="/resources/admission2.png" id="toggle2img" class="admission2" />

  </div>

  <div class="footer">
   <div class="fcontainer">
    <div class="k-icon">
     <img src="/resources/KMITL_Logo.png" class="kmitl" />
    </div>
    <div class="social">
     <ul class="wrapper">
      <li class="icon facebook">
       <a
        href="https://www.facebook.com/profile.php?id=100066929134562"
        target="_blank"
        class="tooltip"
        >Facebook<i class="fab fa-facebook-f"></i
       ></a>
      </li>
      <li class="icon twitter">
```

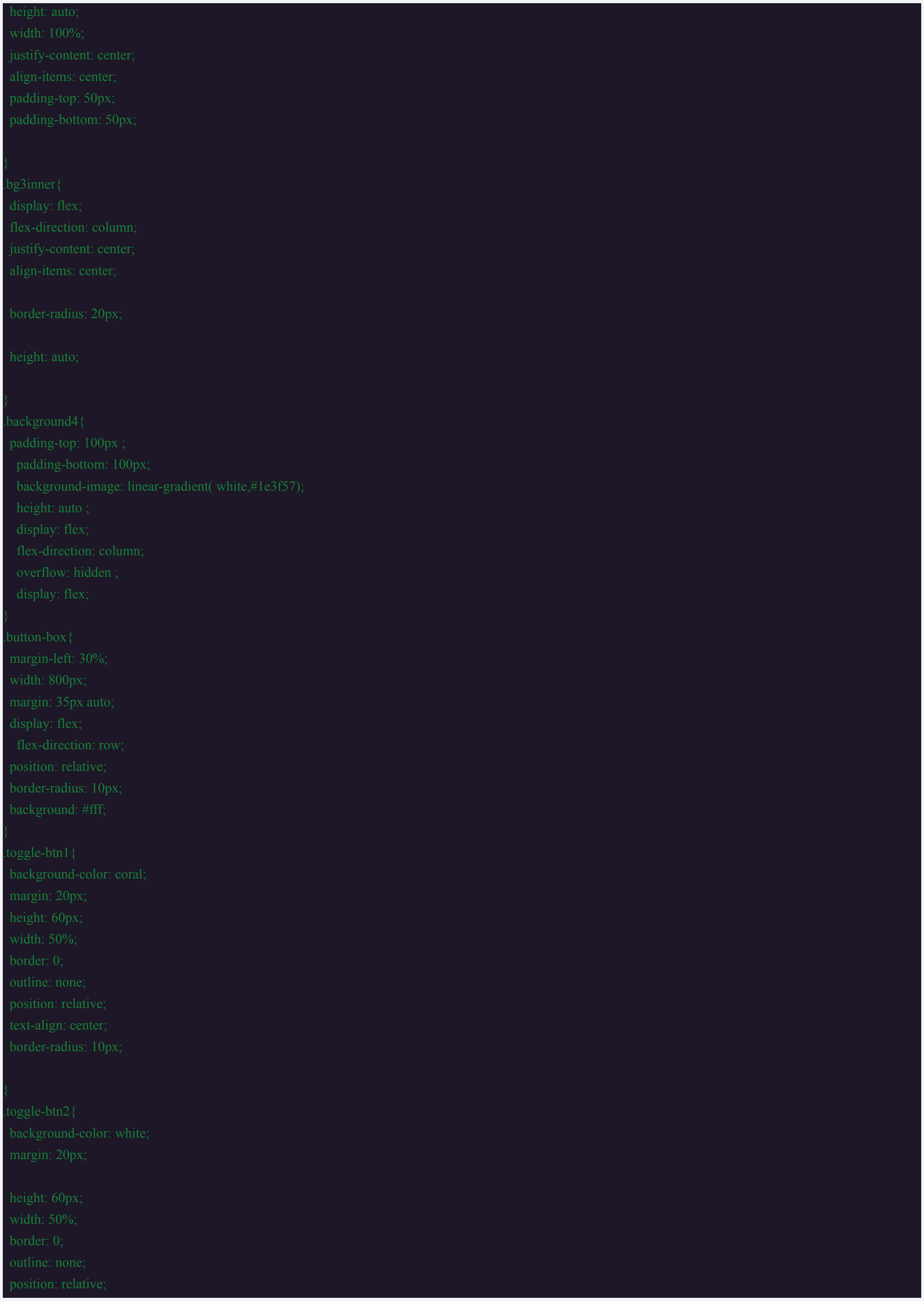
```
       <a href="https://www.twitter.com" target="_blank" class="tooltip"
        >Twitter<i class="fab fa-twitter"></i
       ></a>
      </li>
      <li class="icon instagram">
       <a
        href="https://www.instagram.com/se_kmitl/"
        target="_blank"
        class="tooltip"
        >Instagram<i class="fab fa-instagram"></i
       ></a>
      </li>
     </ul>
     <p>
      © 2023 Software Engineering, King Mongkut's Institute of Technology
      Ladkrabang
     </p>
    </div>
   </div>
  </div>
  <script src="/static/se.js"></script>
 </body>
</html>
```

Se.css

Unset

```
html{
 font-family: monospace !important;
 scroll-behavior: smooth;

}


 a{
  text-decoration: none;

}
/*
a:hover{
  background-color: #451952 !important;
  box-shadow: 10px 10px 0px #1D1A39 !important;
  top: +5px !important;
  left: +5px !important;
} */
.tag{
 background-color: #1e3f57 !important;
}
.kmitl{
  margin-top: 60px;
  margin-left: 60px;
  width: 200px;
}
```

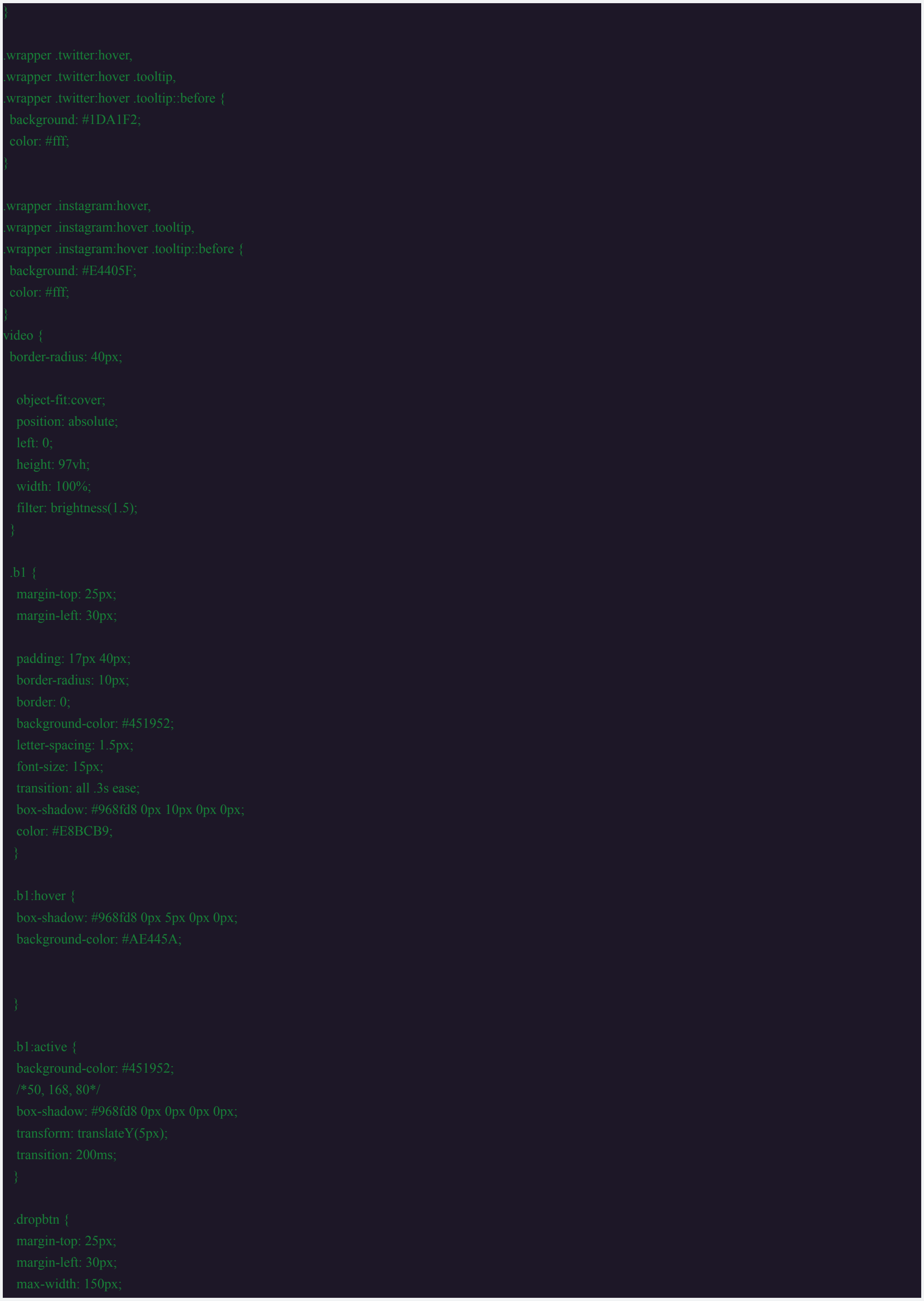
```
.background1{

    background-size: cover;
    background-repeat: no-repeat;
    background-position: center;
    width: 100%;
    height: 100vh;
    display: flex;
    justify-content: flex-start;
    background-image: linear-gradient(#1e3f57, white);
}
.background2{
    padding-top: 100px ;
    padding-bottom: 100px;
    padding-left: 5%;
    background-image: linear-gradient(#1e3f57, white);
    height: 80vh ;
    overflow: hidden ;
    display: flex;
}
.bg2head{
    margin-left: 40%;
    margin-top: 30px;
    padding-bottom: 20px;
    color: #cedce7;
    width: 100%;
    font-size: 3em;
    font-weight: bold;
}
.background3{

  padding-left: 5%;
  padding-right: 5%;
  padding-top: 80px;
  padding-bottom: 300px;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center;
  text-align: center;
  height: auto;
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  background-image: linear-gradient( white ,#1e3f57);
}
.bg3head{
  padding-top: 30px;
  margin-left: 30%;
  margin-right: 30%;
  border-radius: 20px;
  background-color: #c9d2d8;
  font-size: 60px;
  font-weight: bold;
  color: #1e3f57;
  padding-bottom: 20px;
  border: solid #1e3f57;
}
.innerbg3{
  display: flex;
  flex-direction: row;
```



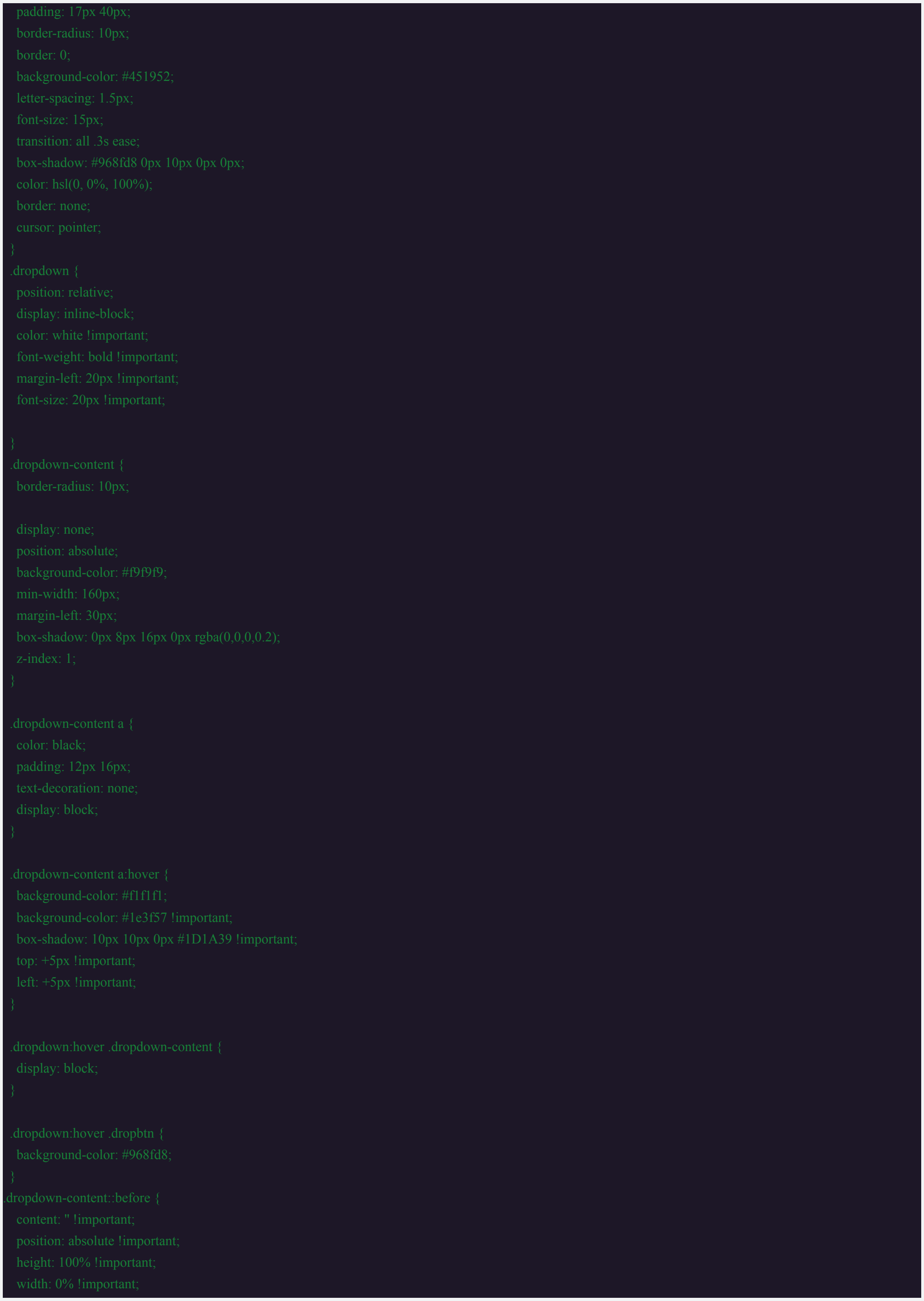
```
  height: auto;
  width: 100%;
  justify-content: center;
  align-items: center;
  padding-top: 50px;
  padding-bottom: 50px;

}
.bg3inner{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;

  border-radius: 20px;

  height: auto;

}
.background4{
  padding-top: 100px ;
    padding-bottom: 100px;
    background-image: linear-gradient( white,#1e3f57);
    height: auto ;
    display: flex;
    flex-direction: column;
    overflow: hidden ;
    display: flex;
}
.button-box{
  margin-left: 30%;
  width: 800px;
  margin: 35px auto;
  display: flex;
    flex-direction: row;
  position: relative;
  border-radius: 10px;
  background: #fff;
}
.toggle-btn1{
  background-color: coral;
  margin: 20px;
  height: 60px;
  width: 50%;
  border: 0;
  outline: none;
  position: relative;
  text-align: center;
  border-radius: 10px;

}
.toggle-btn2{
  background-color: white;
  margin: 20px;

  height: 60px;
  width: 50%;
  border: 0;
  outline: none;
  position: relative;
```

```
 text-align: center;
 border-radius: 10px;

}

.admission1{
 width: 80%;
 height: auto;
 margin-left: 10%;
 margin-right: 10%;

}
.admission2{
 margin-top:140px;
 width: 80%;
 height: auto;
 position: absolute;
 margin-left: 10%;
 margin-right: 10%;
 visibility: hidden;

}
.footer {

 height: 200px;
 width: 100%;
 background-color: #1e3f57;
 color: white;
 text-align: center;
}
.fcontainer{
 display: flex;
 height:200px;
}
.k-icon{
 width: 400px;
 border-right: 2px solid white;
}
.instagram{
 background-color: #E4405F !important;
}
.twitter{
 background-color: #1DA1F2 !important;
}
.facebook{
 background-color: #1877F2 !important;
}
.social{
 margin-left: 100px;
}
.wrapper {
 display: inline-flex;
 list-style: none;
 height: 100px;
 width: 100%;
 padding-top: 20px;
 font-family: "Poppins", sans-serif;
 justify-content: center;
}
```

```
.wrapper .icon {
 position: relative;
 background: #fff;
 border-radius: 50%;
 margin: 10px;
 width: 50px;
 height: 50px;
 font-size: 18px;
 display: flex;
 justify-content: center;
 align-items: center;
 flex-direction: column;
 box-shadow: 0 10px 10px rgba(0, 0, 0, 0.1);
 cursor: pointer;
 transition: all 0.2s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}

.wrapper .tooltip {
 position: absolute;
 top: 0;
 font-size: 14px;
 background: #fff;
 color: #fff;
 padding: 5px 8px;
 border-radius: 5px;
 box-shadow: 0 10px 10px rgba(0, 0, 0, 0.1);
 opacity: 0;
 pointer-events: none;
 transition: all 0.3s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}

.wrapper .tooltip::before {
 position: absolute;
 content: "";
 height: 8px;
 width: 8px;
 background: #fff;
 bottom: -3px;
 left: 50%;
 transform: translate(-50%) rotate(45deg);
 transition: all 0.3s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}

.wrapper .icon:hover .tooltip {
 top: -15px ;
 opacity: 1;
 visibility: visible;
 pointer-events: auto;
}

.wrapper .icon:hover span,
.wrapper .icon:hover .tooltip {
 text-shadow: 0px -1px 0px rgba(0, 0, 0, 0.1);
}

.wrapper .facebook:hover,
.wrapper .facebook:hover .tooltip,
.wrapper .facebook:hover .tooltip::before {
 background: #1877F2;
 color: #fff;
```

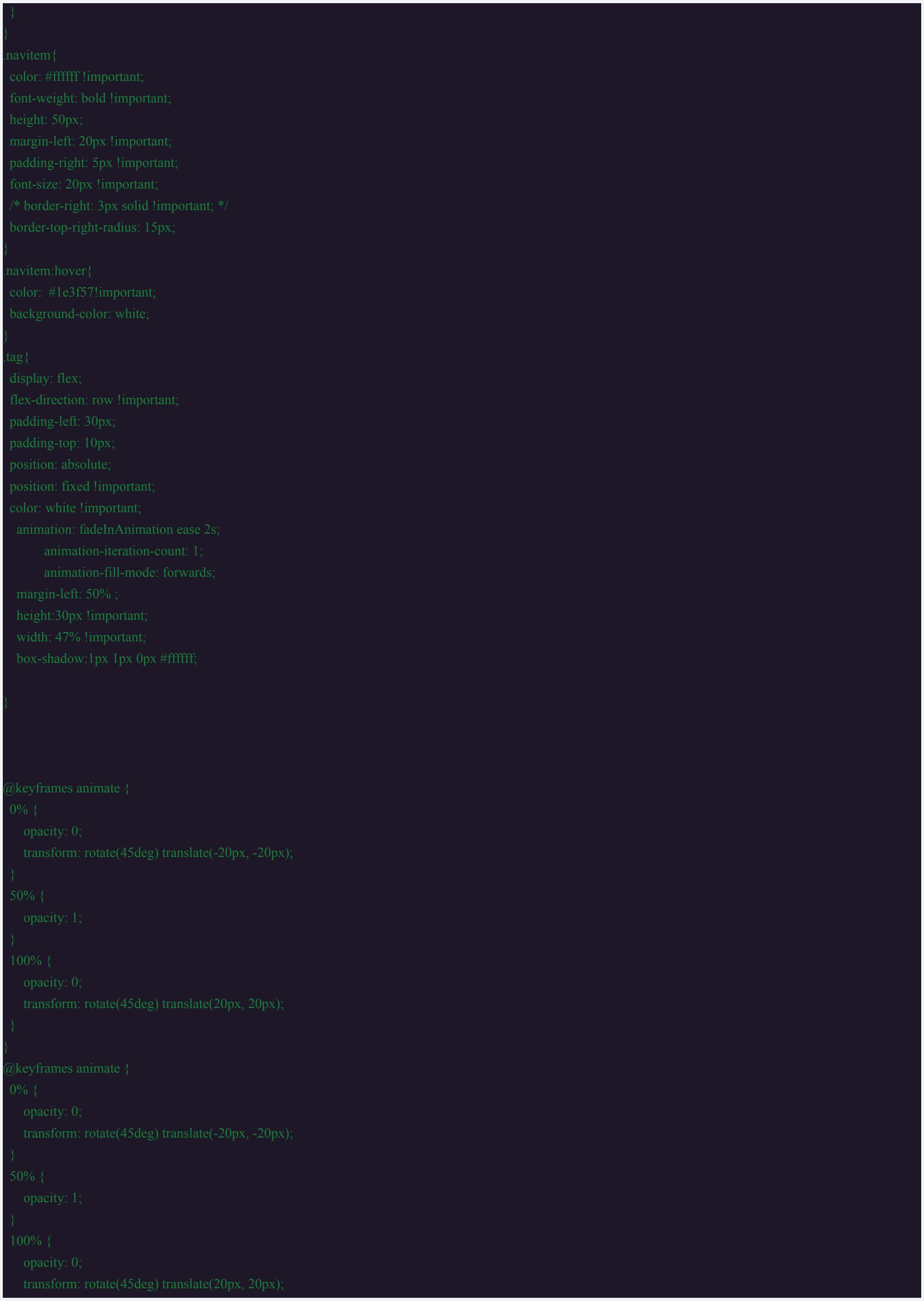


```
}

.wrapper .twitter:hover,
.wrapper .twitter:hover .tooltip,
.wrapper .twitter:hover .tooltip::before {
 background: #1DA1F2;
 color: #fff;
}

.wrapper .instagram:hover,
.wrapper .instagram:hover .tooltip,
.wrapper .instagram:hover .tooltip::before {
 background: #E4405F;
 color: #fff;
}
video {
 border-radius: 40px;

   object-fit:cover;
   position: absolute;
   left: 0;
   height: 97vh;
   width: 100%;
   filter: brightness(1.5);
 }

 .b1 {
   margin-top: 25px;
   margin-left: 30px;

   padding: 17px 40px;
   border-radius: 10px;
   border: 0;
   background-color: #451952;
   letter-spacing: 1.5px;
   font-size: 15px;
   transition: all .3s ease;
   box-shadow: #968fd8 0px 10px 0px 0px;
   color: #E8BCB9;
  }

  .b1:hover {
   box-shadow: #968fd8 0px 5px 0px 0px;
   background-color: #AE445A;


  }

  .b1:active {
   background-color: #451952;
   /*50, 168, 80*/
   box-shadow: #968fd8 0px 0px 0px 0px;
   transform: translateY(5px);
   transition: 200ms;
  }

  .dropbtn {
   margin-top: 25px;
   margin-left: 30px;
   max-width: 150px;
```

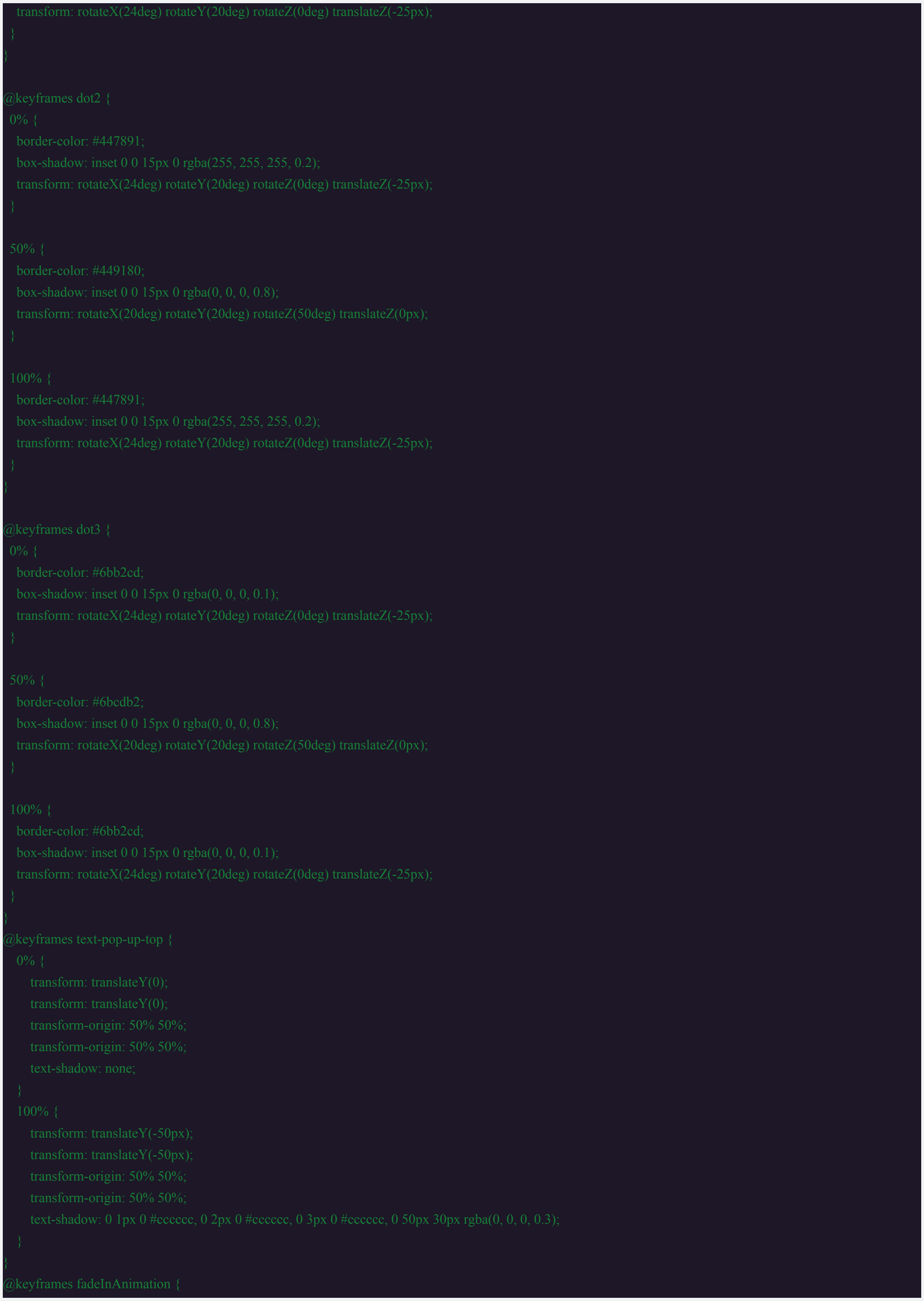


```
  padding: 17px 40px;
  border-radius: 10px;
  border: 0;
  background-color: #451952;
  letter-spacing: 1.5px;
  font-size: 15px;
  transition: all .3s ease;
  box-shadow: #968fd8 0px 10px 0px 0px;
  color: hsl(0, 0%, 100%);
  border: none;
  cursor: pointer;
 }
 .dropdown {
  position: relative;
  display: inline-block;
  color: white !important;
  font-weight: bold !important;
  margin-left: 20px !important;
  font-size: 20px !important;

 }
 .dropdown-content {
  border-radius: 10px;

  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  margin-left: 30px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
 }

 .dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
 }

 .dropdown-content a:hover {
  background-color: #f1f1f1;
  background-color: #1e3f57 !important;
  box-shadow: 10px 10px 0px #1D1A39 !important;
  top: +5px !important;
  left: +5px !important;
 }

 .dropdown:hover .dropdown-content {
  display: block;
 }

 .dropdown:hover .dropbtn {
  background-color: #968fd8;
 }
.dropdown-content::before {
  content: " !important;
  position: absolute !important;
  height: 100% !important;
  width: 0% !important;
```

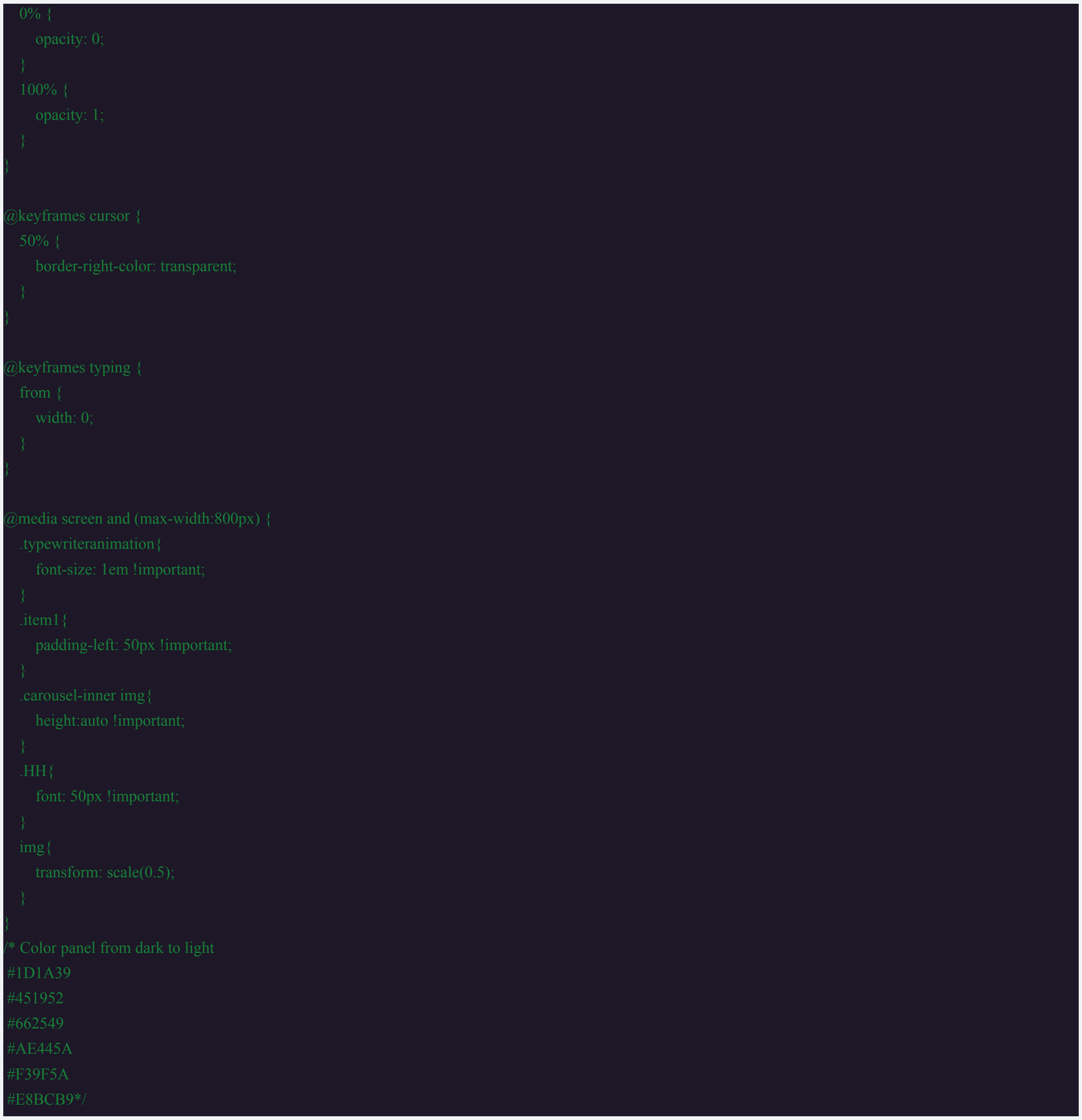
```
    top: 0 !important;
    left: -40px !important;
    transform: skewX(45deg) !important;
    background-color: #306e9b !important;
    z-index: -1 !important;
    transition: all 1s !important;
  }

.dropdown-content:hover::before {
    width: 160%;
}
.item1{
    animation: fadeInAnimation ease 3s;
            animation-iteration-count: 1;
            animation-fill-mode: forwards;
    padding-left: 60px !important;
    position: relative !important;
    padding-top: 50px !important;
    width: 10% !important;
    flex-shrink: 0 !important;
    position: absolute !important;
}

.item2{
    margin-left: 25% !important;
    color: #1e3f57 !important;
    font-size: 40px;

}
.item3{
  margin-left: 50px !important;
}
.item4{
  margin-top: 50px !important;
  margin-left: 100px !important;
}
.item3 a{
  font-size: 3em !important;
  background-color: #140e50;
  border-radius: 20px;
}
.item4 a{
  font-size: 3em !important;
  background-color: #140e50;
  border-radius: 20px;
}
.se {
  width: 200px;
  height: auto;
  filter: brightness(0.2);

  /* -webkit-animation: se 1.5s ease-in-out infinite both;
          animation: se 1.5s ease-in-out infinite both; */

  /* border: #140e50 1px solid; */
}
#login-div{
  width: 200px;
  border-radius: 25px;
  background-color: white;
  box-shadow: 0px 0px 10px white;
}
#login-text{
  text-align: center;
  padding-bottom: 5%;
  font-family: monospace;
  font-weight: bold;
  letter-spacing: 2px;
  color: rgb(47, 90, 88);
}
.item5{
  margin-top:30px;

}
.accordion {
  border-radius: 10px;
  border: #140e50 2px solid !important;
  background-color: #e4d4d3;
  color: #1D1A39;
  cursor: pointer;
  font-size: 20px;
  font-weight: bold;
  padding: 18px;
  width: 500px !important;
  height: 80px;
  width: 100% !important;
  text-align: center !important;
  border: none;
  outline: none;
  transition: background-color 0.4s;
}

.active, .accordion:hover {
  color: #e4d4d3;
  height: auto;
  background-color: #1e3f57 !important;
    box-shadow: 10px 10px 0px #1D1A39 !important;
    top: +5px !important;
    left: +5px !important;
}

.panel {
  margin-top: 10px;
  border-radius: 10px;
  padding: 0 18px;
  max-width: 670px;
  background-color: #e5d8db;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  display: none;
  overflow: hidden;
  font-size: 15px;
  text-align: justify;
  transition: max-height 0.2s ease-out;
}
@-webkit-keyframes se {
  from {
    -webkit-transform: scale(1);
            transform: scale(1);
    -webkit-transform-origin: center center;
            transform-origin: center center;
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
  }
  10% {
    -webkit-transform: scale(0.91);
            transform: scale(0.91);
    -webkit-animation-timing-function: ease-in;
            animation-timing-function: ease-in;
  }
  17% {
    -webkit-transform: scale(0.98);
            transform: scale(0.98);
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
  }
  33% {
    -webkit-transform: scale(0.87);
            transform: scale(0.87);
    -webkit-animation-timing-function: ease-in;
            animation-timing-function: ease-in;
  }
  45% {
    -webkit-transform: scale(1);
            transform: scale(1);
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
  }
}
@keyframes se {
  from {
    -webkit-transform: scale(1);
            transform: scale(1);
    -webkit-transform-origin: center center;
            transform-origin: center center;
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
  }
  10% {
    -webkit-transform: scale(0.91);
            transform: scale(0.91);
    -webkit-animation-timing-function: ease-in;
            animation-timing-function: ease-in;
  }
  17% {
    -webkit-transform: scale(0.98);
            transform: scale(0.98);
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
  }
  33% {
    -webkit-transform: scale(0.87);
            transform: scale(0.87);
    -webkit-animation-timing-function: ease-in;
            animation-timing-function: ease-in;
  }
  45% {
    -webkit-transform: scale(1);
            transform: scale(1);
    -webkit-animation-timing-function: ease-out;
            animation-timing-function: ease-out;
```



```
  }
}
.navitem{
  color: #ffffff !important;
  font-weight: bold !important;
  height: 50px;
  margin-left: 20px !important;
  padding-right: 5px !important;
  font-size: 20px !important;
  /* border-right: 3px solid !important; */
  border-top-right-radius: 15px;
}
.navitem:hover{
  color:  #1e3f57!important;
  background-color: white;
}
.tag{
  display: flex;
  flex-direction: row !important;
  padding-left: 30px;
  padding-top: 10px;
  position: absolute;
  position: fixed !important;
  color: white !important;
    animation: fadeInAnimation ease 2s;
            animation-iteration-count: 1;
            animation-fill-mode: forwards;
    margin-left: 50% ;
    height:30px !important;
    width: 47% !important;
    box-shadow:1px 1px 0px #ffffff;

}



@keyframes animate {
  0% {
      opacity: 0;
      transform: rotate(45deg) translate(-20px, -20px);
  }
  50% {
      opacity: 1;
  }
  100% {
      opacity: 0;
      transform: rotate(45deg) translate(20px, 20px);
  }
}
@keyframes animate {
  0% {
      opacity: 0;
      transform: rotate(45deg) translate(-20px, -20px);
  }
  50% {
      opacity: 1;
  }
  100% {
      opacity: 0;
      transform: rotate(45deg) translate(20px, 20px);
  }
}

.typewriteranimation1 {
    margin-top: 20%;
    padding-left: 450px !important;
    width: 250px;
    margin-left: 150px;
    font-size: 2em;
    letter-spacing: 10px;
    font-family: monospace;
    border-right: 5px solid;
    white-space: nowrap;
    overflow: hidden;
    flex-shrink: 0 ;

    animation: typing 4s steps(25),
        cursor 0.5s step-end infinite alternate;
}
.typewriteranimation2 {
  width: 950px;
  margin-top: 200px;
  margin-left: 350px;
  font-size: 10em;
  letter-spacing: 5px;
  color: #1e3f57;
  font-family: monospace;
  border-right: 2px solid;
  white-space: nowrap;
  overflow: hidden;
  flex-shrink: 0 ;

  animation: typing 4s steps(25),
      cursor 0.5s step-end infinite alternate;
}

.text-pop-up-top {
    padding-top: 100px !important;
    color: #a56c77 !important;
  animation: text-pop-up-top 0.5s cubic-bezier(0.250, 0.460, 0.450, 0.940) both;
  animation: text-pop-up-top 0.5s cubic-bezier(0.250, 0.460, 0.450, 0.940) both;
}
.HH{
    font-size: 3em !important;
    color: rgb(176, 55, 176);
    text-transform: uppercase;
    text-decoration: none;
    padding: 10px 20px;
    font-size: 17px;
    font-weight: bold;
    background: transparent;
    position: relative;
    transition: all 1s;
    overflow: hidden;
}
.HH:hover{
    color: white;
    background-color: #451952 !important;
    box-shadow: 10px 10px 0px #1D1A39 !important;
    top: +5px !important;
    left: +5px !important;
}
.HH::before {
    content: '';
    position: absolute;
    height: 100%;
    width: 0%;
    top: 0;
    left: -40px;
    transform: skewX(45deg);
    background-color: purple;
    z-index: -1;
    transition: all 1s;
  }

.HH:hover::before {
    width: 160%;
}
.HHH{
  font-size: 20px !important;
  color: #e1b7c0;
  text-transform: uppercase;
  text-decoration: none;
  padding: 10px 20px;
  font-size: 17px;
  font-weight: bold;
  background: transparent;
  position: relative;
  transition: all 1s;
  overflow: hidden;
}
.HHH:hover{
  border-radius: 20px;
  color: white;
  background-color: #1e3f57!important;
    box-shadow: 10px 10px 0px #1D1A39 !important;
    top: +5px !important;
    left: +5px !important;
}
.HHH::before {
  content: '';
  position: absolute;
  height: 100%;
  width: 0%;
  top: 0;
  left: -40px;
  transform: skewX(45deg);
  background-color: #1e3f57;
  z-index: -1;
  transition: all 1s;
}

.HHH:hover::before {
  width: 160%;
}

.card1 {
  width: 600px;
  height: 400px;
  margin-top: 100px !important;
  border-radius: 20px;
  background: #19354a;
  display: flex;
  position: absolute;
  gap: 5px;
  padding: .4em;
}

.card1 p {
  height: 100%;
  flex: 1;
  overflow: hidden;
  cursor: pointer;
  border-radius: 2px;
  transition: all .5s;
  background: transparent;
  display: flex;
  justify-content: center;
  align-items: center;
}

.card1 p:hover {
  flex: 30;
}

.card1 p img {
  min-width: 14em;
  padding: .5em;
  text-align: center;
  transform: rotate(-90deg);
  transition: all .5s;
  text-transform: uppercase;
  color: #1D1A39;
  letter-spacing: .1em;
}

.card1 p:hover img{
  transform: rotate(0);
}

.card2 {
  width: 600px;
  margin-left: 50%;
  height: 400px;
  margin-top: 100px !important;
  border-radius: 20px;
  background: #19354a;
  display: flex;
  position: absolute;
  gap: 5px;
  padding: .4em;
}

.card2 p {
  height: 100%;
  flex: 1;
  overflow: hidden;
  cursor: pointer;
  border-radius: 2px;
  transition: all .5s;
  background: transparent;
  display: flex;
  justify-content: center;
  align-items: center;
}

.card2 p:hover {
  flex: 30;
}

.card2 p img {
  min-width: 14em;
  padding: .5em;
  text-align: center;
  transform: rotate(-90deg);
  transition: all .5s;
  text-transform: uppercase;
  color: #1D1A39;
  letter-spacing: .1em;
}

.card2 p:hover img{
  transform: rotate(0);
}
/* The loader container */
.loader1 {
  margin-top: 100px;
  width: 100%;
  height: auto;
  perspective: 200px;
  z-index:9999 !important;
  position: absolute;

}


/* The dot */
.dot {
  position: absolute;
  top: 50%;
  left: 50%;
  width: 120px;
  height: 120px;
  margin-top: -60px;
  margin-left: -60px;
  border-radius: 100px;
  border: 40px outset #1e3f57;
  transform-origin: 50% 50%;
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  background-color: transparent;
  animation: dot1 1000ms cubic-bezier(.49,.06,.43,.85) infinite;
}

.dot:nth-child(2) {
  width: 140px;
  height: 140px;
  margin-top: -70px;
  margin-left: -70px;
  border-width: 30px;
  border-color: #447891;
  animation-name: dot2;
  animation-delay: 75ms;
  box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}

.dot:nth-child(3) {
  width: 160px;
  height: 160px;
  margin-top: -80px;
  margin-left: -80px;
  border-width: 20px;
  border-color: #6bb2cd;
  animation-name: dot3;
  animation-delay: 150ms;
  box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}
.loader {
    position: fixed;
    top: 0;
    left: 0;
    width: 100vw;
    height: 100vh;
    display: flex;
    align-items: center;
    justify-content: center;
    background: #333333;
    transition: opacity 2s, visibility 2s;
  }

  .loader--hidden {
    opacity: 0;
    visibility: hidden;
  }



  @keyframes loading {
    from {
      transform: rotate(0turn);
    }
    to {
      transform: rotate(1turn);
    }
  }

@keyframes dot1 {
  0% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #1e574f;
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #1e3f57;
```



```
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot2 {
  0% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #449180;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot3 {
  0% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #6bcdb2;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}
@keyframes text-pop-up-top {
    0% {
        transform: translateY(0);
        transform: translateY(0);
        transform-origin: 50% 50%;
        transform-origin: 50% 50%;
        text-shadow: none;
    }
    100% {
        transform: translateY(-50px);
        transform: translateY(-50px);
        transform-origin: 50% 50%;
        transform-origin: 50% 50%;
        text-shadow: 0 1px 0 #cccccc, 0 2px 0 #cccccc, 0 3px 0 #cccccc, 0 50px 30px rgba(0, 0, 0, 0.3);
    }
}
@keyframes fadeInAnimation {
```

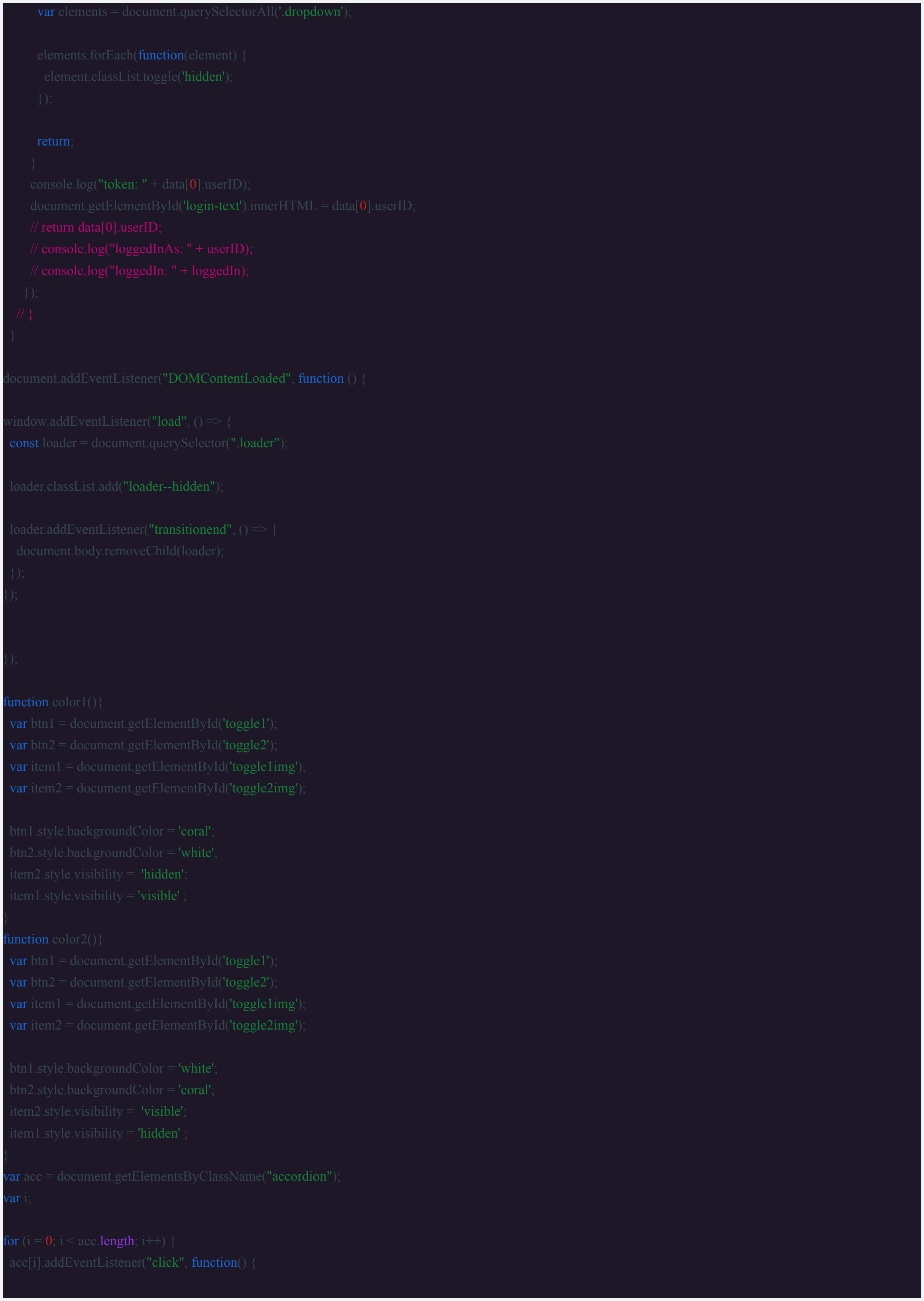


```
    0% {
        opacity: 0;
    }
    100% {
        opacity: 1;
    }
}

@keyframes cursor {
    50% {
        border-right-color: transparent;
    }
}

@keyframes typing {
    from {
        width: 0;
    }
}

@media screen and (max-width:800px) {
    .typewriteranimation{
        font-size: 1em !important;
    }
    .item1{
        padding-left: 50px !important;
    }
    .carousel-inner img{
        height:auto !important;
    }
    .HH{
        font: 50px !important;
    }
    img{
        transform: scale(0.5);
    }
}
/* Color panel from dark to light
 #1D1A39
 #451952
 #662549
 #AE445A
 #F39F5A
 #E8BCB9*/
```

Se.js

JavaScript

```
var userID = 0;

window.onload = function() {
  fetchUserID();

  document.getElementById('login-div').addEventListener('click', () => {
    window.location.href = `login.html`;
  });

}

function fetchUserID() {
  // if (userID == 0) {
    fetch('http://localhost:8000/tokenGet')
      .then(response => response.json())
      .then(data => {
        if (data.length == 0) {
          console.log("no token");
          document.getElementById('login-text').innerHTML = "Login";

          console.log("loggedInAs: " + document.getElementById('login-text').innerHTML);
          var elements = document.querySelectorAll('.dropdown');

          elements.forEach(function(element) {
            element.style.visibility = 'hidden';
          });

          return;
        }
        var expire = data[0].expire;
        let now = new Date(); // Get the current date and time
        let hours = now.getHours(); // Get the hours component
        let minutes = now.getMinutes(); // Get the minutes component

        let nowInMinutes = hours * 60 + minutes; // Convert hours and minutes to total minutes
        // console.log("userID: " + data[0].userID);
        console.log("expire: " + expire);
        console.log("nowInMinutes: " + nowInMinutes);
        if (nowInMinutes > expire) {

          fetch('http://localhost:8000/expiredToken', {
            method: 'post',
            headers: {
              'Content-Type': 'application/json',
            },

            body: JSON.stringify({
              "userID": data[0].userID
            })
          });

          console.log("token expired");
          document.getElementById('login-text').innerHTML = "Login";

          console.log("loggedInAs: " + document.getElementById('login-text').innerHTML);
```

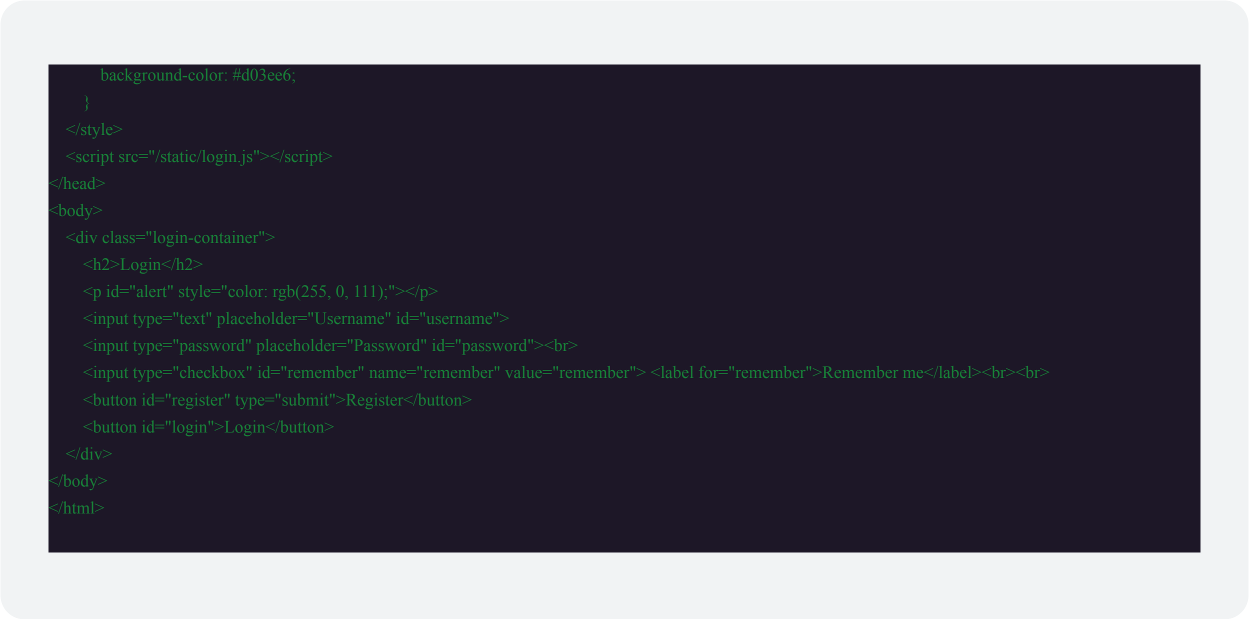


```
          var elements = document.querySelectorAll('.dropdown');

          elements.forEach(function(element) {
            element.classList.toggle('hidden');
          });

          return;
        }
        console.log("token: " + data[0].userID);
        document.getElementById('login-text').innerHTML = data[0].userID;
        // return data[0].userID;
        // console.log("loggedInAs: " + userID);
        // console.log("loggedIn: " + loggedIn);
      });
    // }
  }

document.addEventListener("DOMContentLoaded", function () {

window.addEventListener("load", () => {
  const loader = document.querySelector(".loader");

  loader.classList.add("loader--hidden");

  loader.addEventListener("transitionend", () => {
    document.body.removeChild(loader);
  });
});


});

function color1(){
  var btn1 = document.getElementById('toggle1');
  var btn2 = document.getElementById('toggle2');
  var item1 = document.getElementById('toggle1img');
  var item2 = document.getElementById('toggle2img');

  btn1.style.backgroundColor = 'coral';
  btn2.style.backgroundColor = 'white';
  item2.style.visibility =  'hidden';
  item1.style.visibility = 'visible' ;
}
function color2(){
  var btn1 = document.getElementById('toggle1');
  var btn2 = document.getElementById('toggle2');
  var item1 = document.getElementById('toggle1img');
  var item2 = document.getElementById('toggle2img');

  btn1.style.backgroundColor = 'white';
  btn2.style.backgroundColor = 'coral';
  item2.style.visibility =  'visible';
  item1.style.visibility = 'hidden' ;
}
var acc = document.getElementsByClassName("accordion");
var i;

for (i = 0; i < acc.length; i++) {
  acc[i].addEventListener("click", function() {
    this.classList.toggle("active");

    var panel = this.nextElementSibling;
    if (panel.style.display === "block") {
      panel.style.display = "none";
    } else {
      panel.style.display = "block";
    }
  });
}
```

Login.html

Unset

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <!-- <meta name="viewport" content="width=device-width, initial-scale=1.0"> -->
    <title>Login Page</title>
    <style>
        body {
            background-color: #dbcbc0;
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }

        .login-container {
            /* background-color: #2ecc71; */
            background-color: white;
            border-radius: 20px;
            box-shadow: 0px 0px 10px rgba(132, 19, 149, 0.3);
            padding: 20px;
            width: 40%;
            height: 35%;
            text-align: center;
        }

        h1 {
            color: #212121;
            font-size: 60px;
        }

        h2 {
            color: #212121;
        }

        input[type="text"],
        input[type="password"] {
            width: 80%;
            padding: 10px;
            margin: 10px 0;
            border: 0.5px solid #bdc3c7;
            border-radius: 5px;
        }

        button {
            background-color: #9900c4;
            color: #fff;
            border: none;
            border-radius: 5px;
            padding: 10px 20px;
            cursor: pointer;
        }

        button:hover {
```

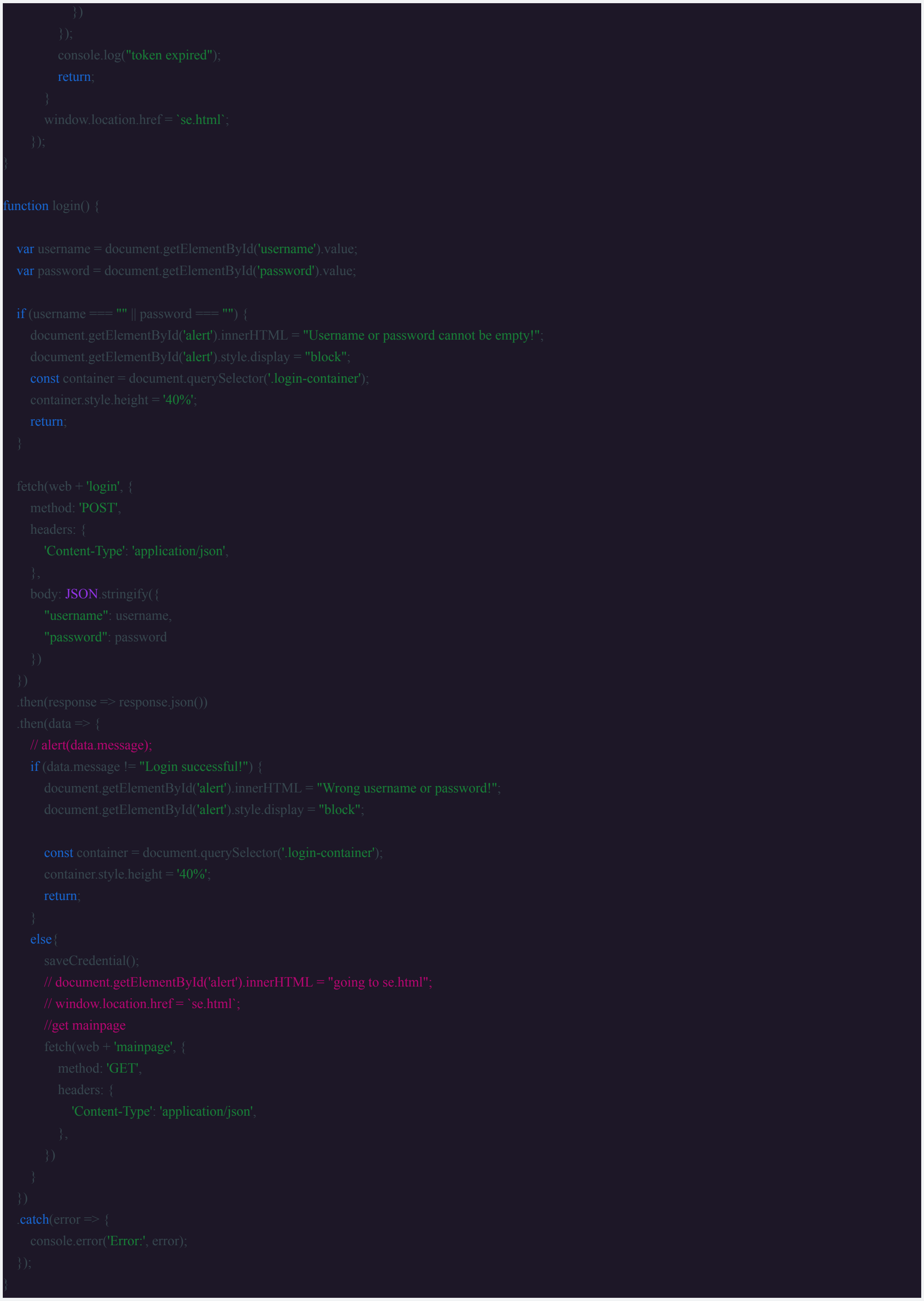


```
            background-color: #d03ee6;
        }
    </style>
    <script src="/static/login.js"></script>
</head>
<body>
    <div class="login-container">
        <h2>Login</h2>
        <p id="alert" style="color: rgb(255, 0, 111);"></p>
        <input type="text" placeholder="Username" id="username">
        <input type="password" placeholder="Password" id="password"><br>
        <input type="checkbox" id="remember" name="remember" value="remember"> <label for="remember">Remember me</label><br><br>
        <button id="register" type="submit">Register</button>
        <button id="login">Login</button>
    </div>
</body>
</html>
```

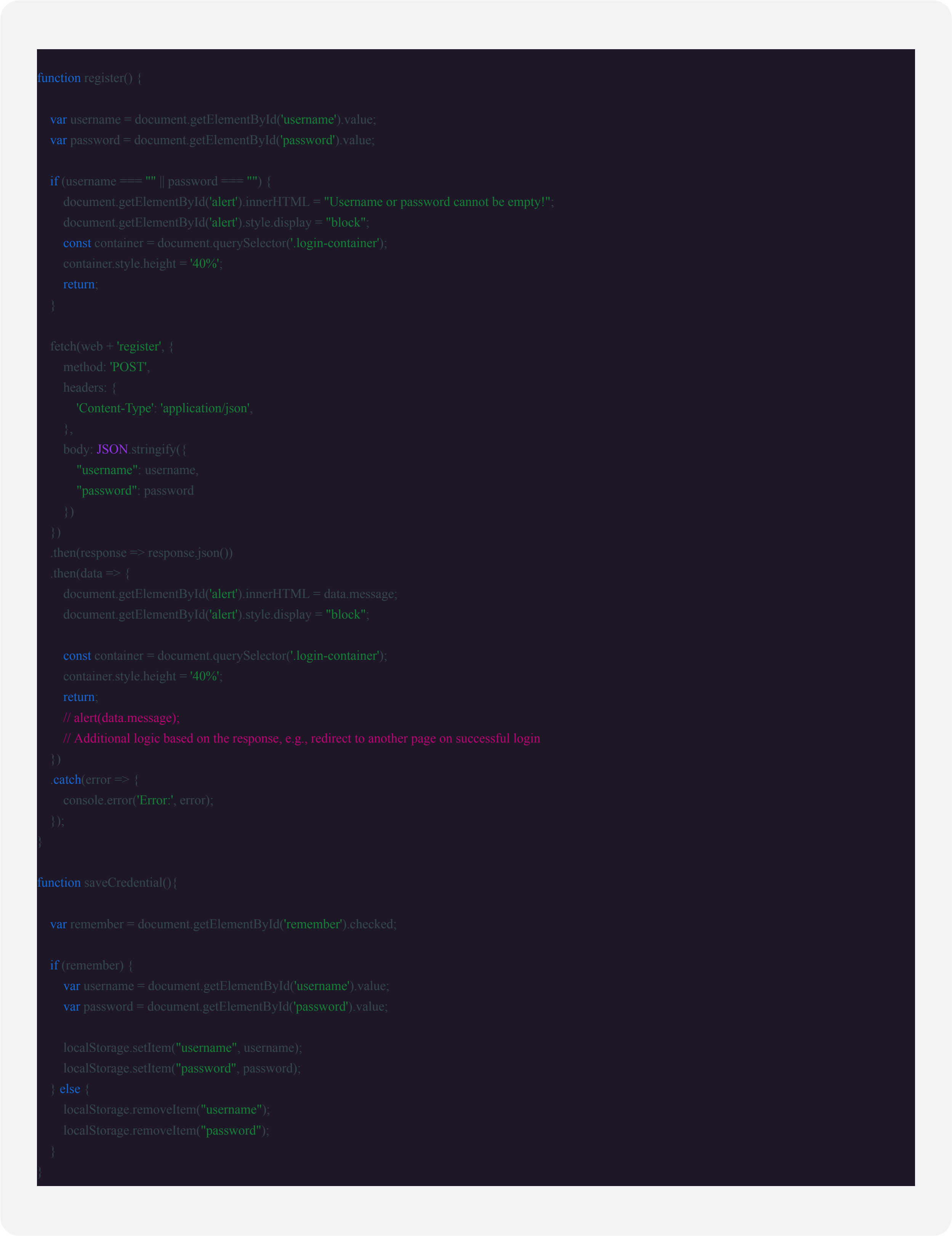
Login.js

JavaScript

```
/*
TODO

-hidden on load for dropdown
-visible when logged in

*/



const web = 'http://localhost:8000/'

window.onload = function() {

    document.getElementById('login').addEventListener('click', login);
    document.getElementById('register').addEventListener('click', register);
    document.getElementById('username').addEventListener('keyup', function(event) {
        if (event.keyCode === 13 && document.getElementById('username').value != "") {
            event.preventDefault();
            document.getElementById('login').click();
        }
    });
    document.getElementById('password').addEventListener('keyup', function(event) {
        if (event.keyCode === 13 && document.getElementById('username').value != "") {
            event.preventDefault();
            document.getElementById('login').click();
        }
    });

    localStorage.getItem("username") ? document.getElementById('username').value = localStorage.getItem("username") : null;
    localStorage.getItem("password") ? document.getElementById('password').value = localStorage.getItem("password") : null;

    document.getElementById('alert').style.display = "none";

    //check if token exists
    fetch(web + 'tokenGet')
        .then(response => response.json())
        .then(data => {
            if (data.length == 0) {
                console.log("no token");
                return;
            }
            var expire = data[0].expire;
            let now = new Date(); // Get the current date and time
            let hours = now.getHours(); // Get the hours component
            let minutes = now.getMinutes(); // Get the minutes component

            let nowInMinutes = hours * 60 + minutes; // Convert hours and minutes to total minutes
            if (nowInMinutes > expire) {
                //delete expired token
                fetch(web + 'expiredToken', {
                    method: 'post',
                    headers: {
                        'Content-Type': 'application/json',
                    },
                    body: JSON.stringify({
                        "userID": data[0].userID
```



```
                    })
                });
                console.log("token expired");
                return;
            }
            window.location.href = `se.html`;
        });
}

function login() {

    var username = document.getElementById('username').value;
    var password = document.getElementById('password').value;

    if (username === "" || password === "") {
        document.getElementById('alert').innerHTML = "Username or password cannot be empty!";
        document.getElementById('alert').style.display = "block";
        const container = document.querySelector('.login-container');
        container.style.height = '40%';
        return;
    }

    fetch(web + 'login', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            "username": username,
            "password": password
        })
    })
    .then(response => response.json())
    .then(data => {
        // alert(data.message);
        if (data.message != "Login successful!") {
            document.getElementById('alert').innerHTML = "Wrong username or password!";
            document.getElementById('alert').style.display = "block";

            const container = document.querySelector('.login-container');
            container.style.height = '40%';
            return;
        }
        else{
            saveCredential();
            // document.getElementById('alert').innerHTML = "going to se.html";
            // window.location.href = `se.html`;
            //get mainpage
            fetch(web + 'mainpage', {
                method: 'GET',
                headers: {
                    'Content-Type': 'application/json',
                },
            })
        }
    })
    .catch(error => {
        console.error('Error:', error);
    });
}
```



```
function register() {

    var username = document.getElementById('username').value;
    var password = document.getElementById('password').value;

    if (username === "" || password === "") {
        document.getElementById('alert').innerHTML = "Username or password cannot be empty!";
        document.getElementById('alert').style.display = "block";
        const container = document.querySelector('.login-container');
        container.style.height = '40%';
        return;
    }

    fetch(web + 'register', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            "username": username,
            "password": password
        })
    })
    .then(response => response.json())
    .then(data => {
        document.getElementById('alert').innerHTML = data.message;
        document.getElementById('alert').style.display = "block";

        const container = document.querySelector('.login-container');
        container.style.height = '40%';
        return;
        // alert(data.message);
        // Additional logic based on the response, e.g., redirect to another page on successful login
    })
    .catch(error => {
        console.error('Error:', error);
    });
}

function saveCredential(){

    var remember = document.getElementById('remember').checked;

    if (remember) {
        var username = document.getElementById('username').value;
        var password = document.getElementById('password').value;

        localStorage.setItem("username", username);
        localStorage.setItem("password", password);
    } else {
        localStorage.removeItem("username");
        localStorage.removeItem("password");
    }
}
```

Blog.html

Unset

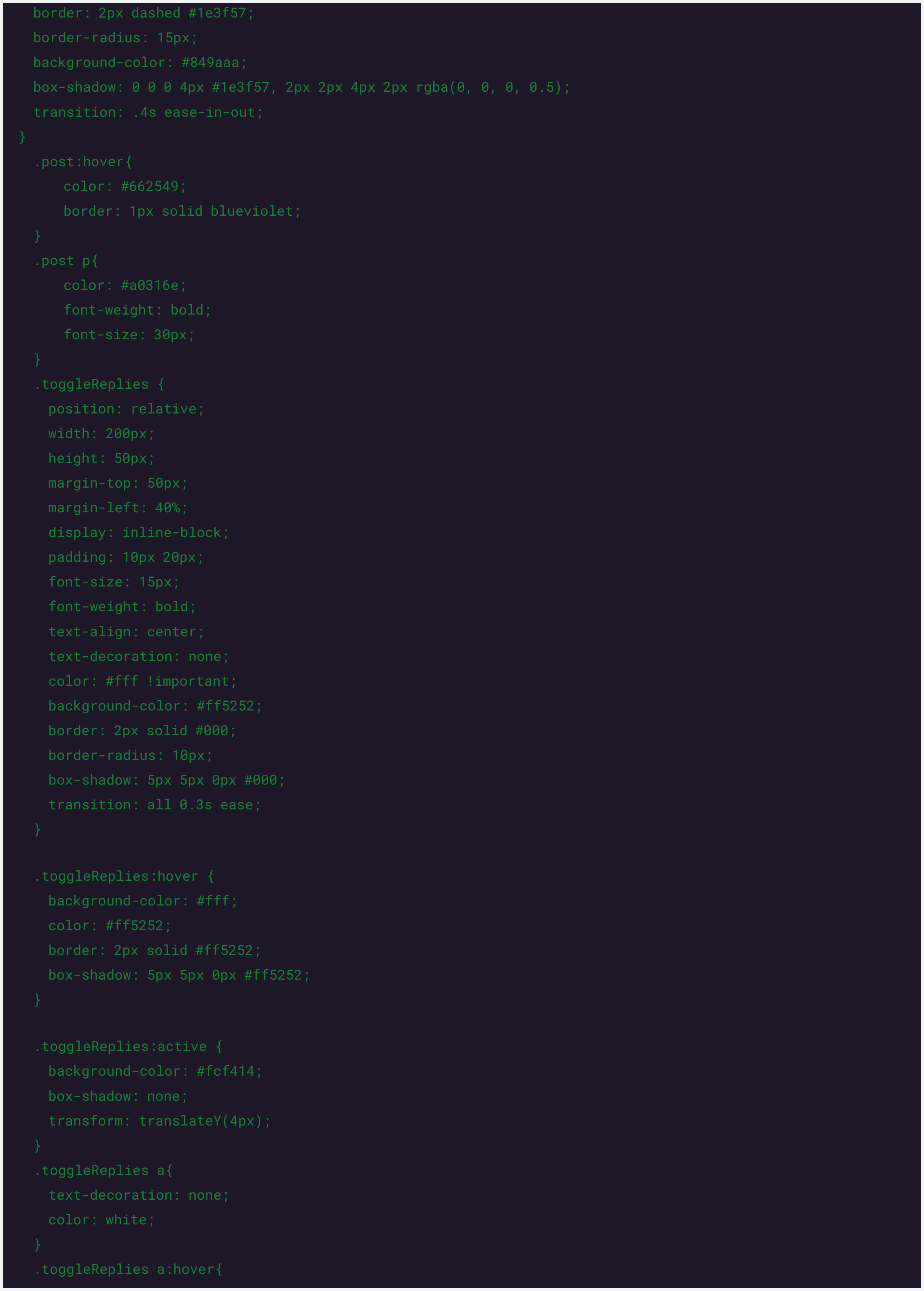
```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <link rel="stylesheet" href="/static/blog.css" />

   <title>My Blog</title>
 </head>
 <body>
   <div class="loader">
     <div class="loader1">
       <div class="dot"></div>
       <div class="dot"></div>
       <div class="dot"></div>
     </div>
   </div>
   <button class="comic-button2"><a href="se.html">Back</a></button>
   <h1 class="title">SE BLOG</h1>
   <input type="text" id="searchInput" placeholder="Search by Title">
   <button id="searchButton">Search</button>
   <form class="form">
     <div id="newPostSection" style="margin-bottom: 20px">
       <p class="heading">Create a New Post</p>
       <input
         type="text"
         id="newPostTitle"
         class="input"
         placeholder="Post Title"
       />
       <br />
       <textarea
         id="newPostContent"
         class="input"
         placeholder="Post Content"
       ></textarea>
       <br />
       <button class="btn" id="newPostButton">Post</button>
     </div>
   </form>
   <!--
       <p class="heading">LOGIN</p>
       <input placeholder="Username" class="input" type="text">
       <input placeholder="Password" class="input" type="password">
       <button class="btn">Submit</button>
   </form> -->
   <div id="blogPosts"></div>

   <script src="/static/blog.js"></script>
 </body>
</html>
```

Blog.css

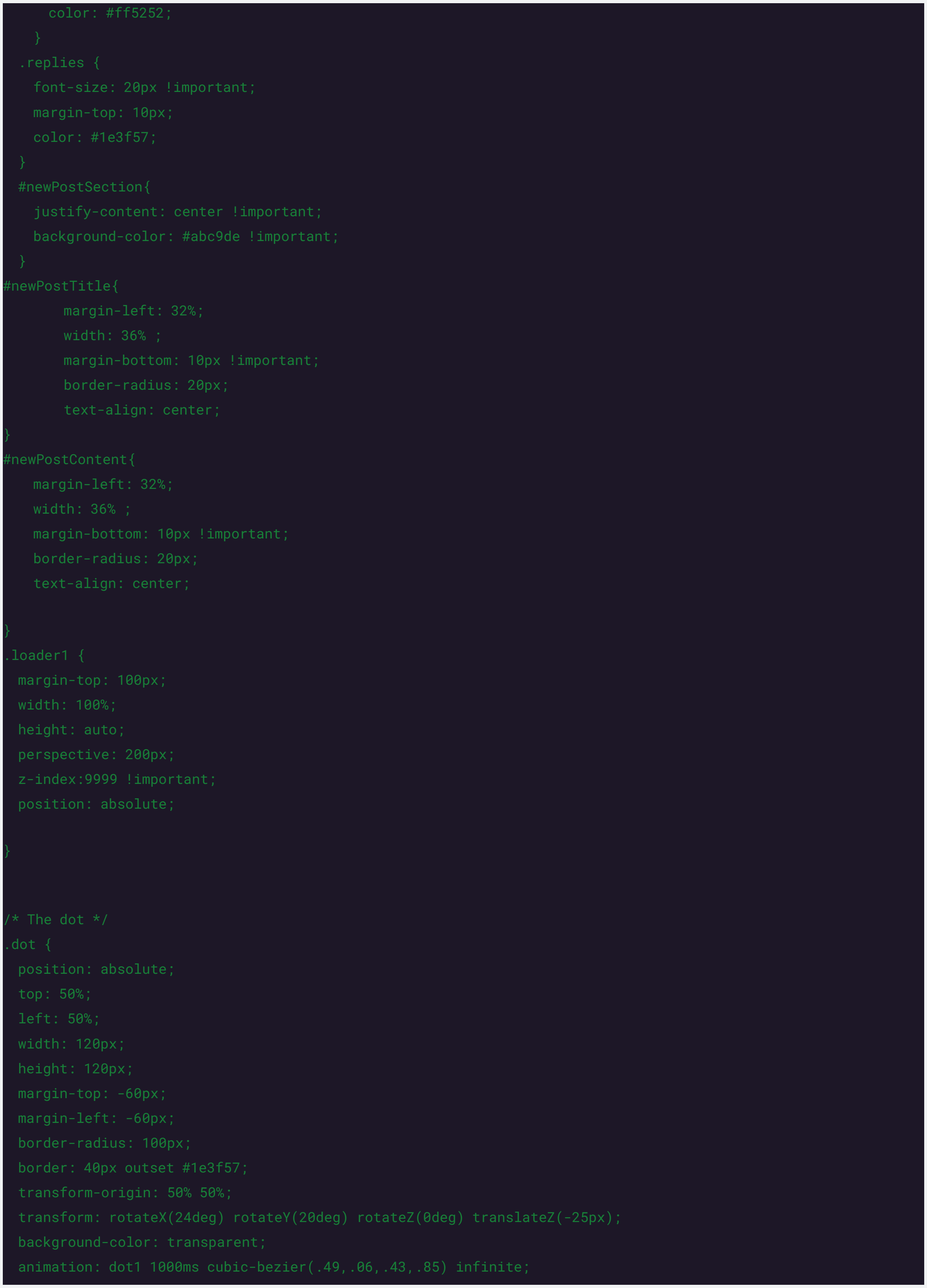
Unset

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 20px;
  }
  html{
    background-color: white;
  }
  h2{
    color: #1e3f57 !important;
  }
  p{
    color: white !important;
  }
  .title{
    width:50%;
    margin-left: 25%;
    text-align: center;
    font-size: 50px;
    color: #1e3f57;
    text-shadow: 2px 2px 2px #662549;
    padding: 1rem 2rem;
 font-weight: 700;
 background: rgb(255, 255, 255);
 color: #1e3f57;
 border-radius: .5rem;
 border-bottom: 2px solid blueviolet;
 border-right: 2px solid blueviolet;
 border-top: 2px solid white;
 border-left: 2px solid white;
 transition-duration: 1s;
 transition-property: border-top,
    border-left,
    border-bottom,
    border-right,
    box-shadow;
  }
  .title:hover{
    color: #662549;
    border-top: 2px solid blueviolet;
 border-left: 2px solid blueviolet;
 border-bottom: 2px solid rgb(238, 103, 238);
 border-right: 2px solid rgb(238, 103, 238);
 box-shadow: rgba(240, 46, 170, 0.4) 5px 5px, rgba(240, 46, 170, 0.3) 10px 10px, rgba(240, 46, 170, 0.2) 15px 15px;
  }
```

```
.comic-button2 {
    position: relative;
    width: 200px;
    height: 40px;
    margin-top: 50px;
    margin-left: 50px;
    display: inline-block;
    padding: 10px 20px;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
  }

  .comic-button2:hover {
    background-color: #fff;
    color: #ff5252;
    border: 2px solid #ff5252;
    box-shadow: 5px 5px 0px #ff5252;
  }

  .comic-button2:active {
    background-color: #fcf414;
    box-shadow: none;
    transform: translateY(4px);
  }
  .comic-button2 a{
    text-decoration: none;
    color: white;
  }
  .comic-button2 a:hover{
    color: #ff5252;
  }
  .post {
    font-weight: bold !important;
    font-size: 30px !important;
    margin-top: 10px;
    margin-bottom: 10px;
    border: 1px solid #ccc;
    display: flex;
    flex-direction: column;
    gap: 10px;
    padding-top: 2em;
    padding-left: 2.8em;
    padding-right: 2.8em;
    padding-bottom: 2.1em;
```

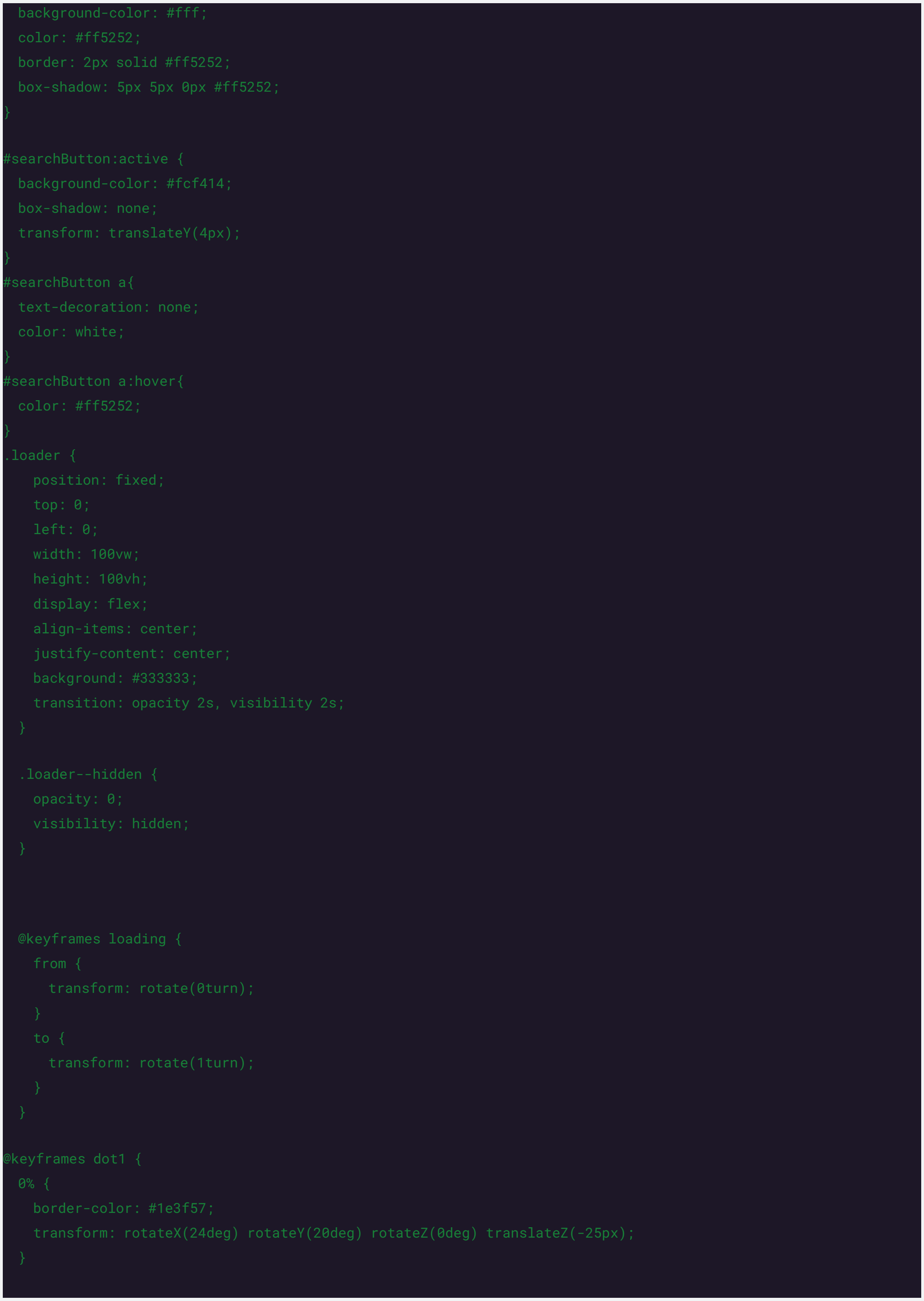


```css
  border: 2px dashed #1e3f57;
  border-radius: 15px;
  background-color: #849aaa;
  box-shadow: 0 0 0 4px #1e3f57, 2px 2px 4px 2px rgba(0, 0, 0, 0.5);
  transition: .4s ease-in-out;
}
  .post:hover{
      color: #662549;
      border: 1px solid blueviolet;
  }
  .post p{
      color: #a0316e;
      font-weight: bold;
      font-size: 30px;
  }
  .toggleReplies {
    position: relative;
    width: 200px;
    height: 50px;
    margin-top: 50px;
    margin-left: 40%;
    display: inline-block;
    padding: 10px 20px;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff !important;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
  }

  .toggleReplies:hover {
    background-color: #fff;
    color: #ff5252;
    border: 2px solid #ff5252;
    box-shadow: 5px 5px 0px #ff5252;
  }

  .toggleReplies:active {
    background-color: #fcf414;
    box-shadow: none;
    transform: translateY(4px);
  }
  .toggleReplies a{
    text-decoration: none;
    color: white;
  }
  .toggleReplies a:hover{
```

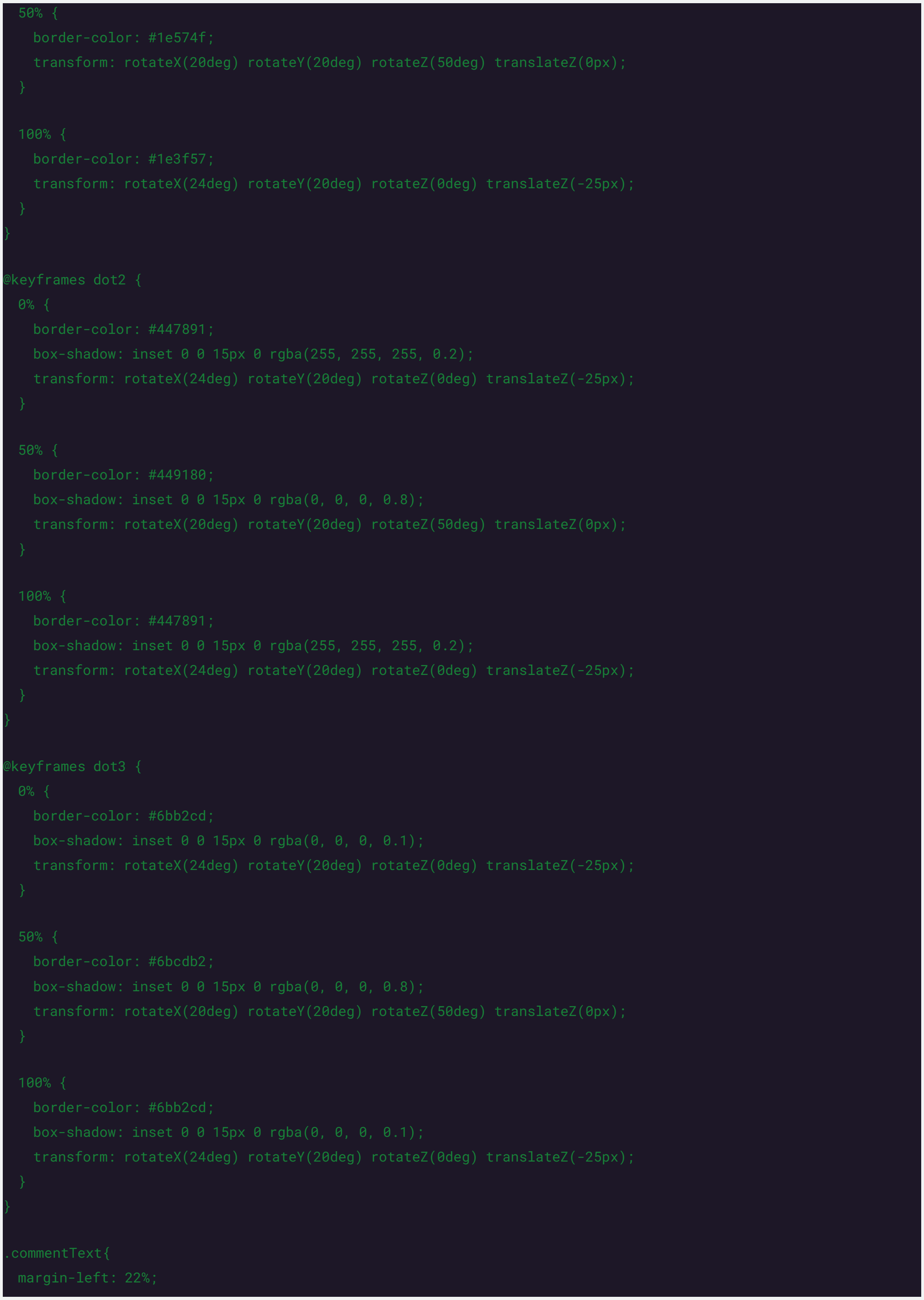


```
      color: #ff5252;
    }
  .replies {
    font-size: 20px !important;
    margin-top: 10px;
    color: #1e3f57;
  }
  #newPostSection{
    justify-content: center !important;
    background-color: #abc9de !important;
  }
#newPostTitle{
       margin-left: 32%;
       width: 36% ;
       margin-bottom: 10px !important;
       border-radius: 20px;
       text-align: center;
}
#newPostContent{
    margin-left: 32%;
    width: 36% ;
    margin-bottom: 10px !important;
    border-radius: 20px;
    text-align: center;

}
.loader1 {
  margin-top: 100px;
  width: 100%;
  height: auto;
  perspective: 200px;
  z-index:9999 !important;
  position: absolute;

}


/* The dot */
.dot {
  position: absolute;
  top: 50%;
  left: 50%;
  width: 120px;
  height: 120px;
  margin-top: -60px;
  margin-left: -60px;
  border-radius: 100px;
  border: 40px outset #1e3f57;
  transform-origin: 50% 50%;
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  background-color: transparent;
  animation: dot1 1000ms cubic-bezier(.49,.06,.43,.85) infinite;
```

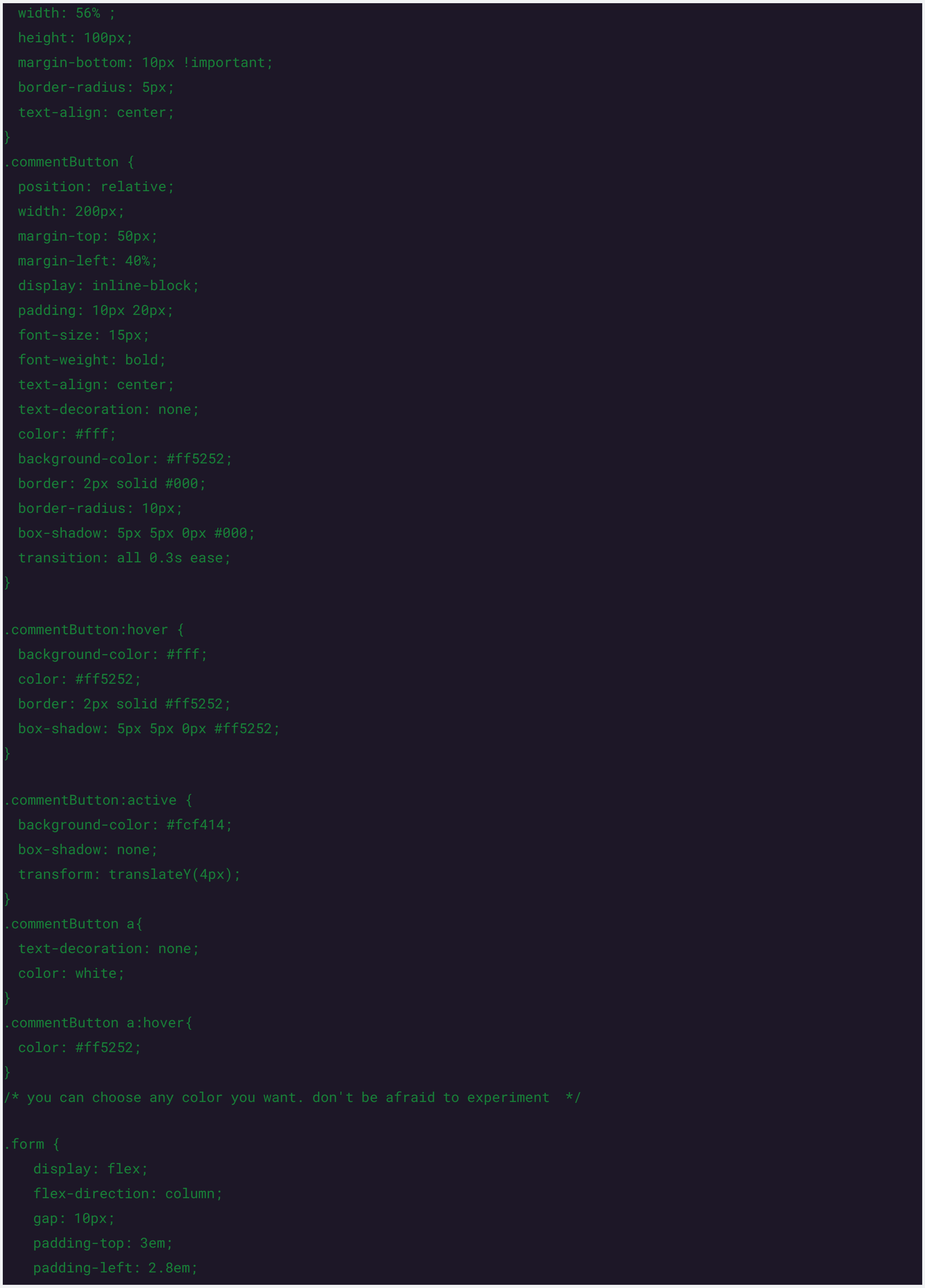
```
}

.dot:nth-child(2) {
  width: 140px;
  height: 140px;
  margin-top: -70px;
  margin-left: -70px;
  border-width: 30px;
  border-color: #447891;
  animation-name: dot2;
  animation-delay: 75ms;
  box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}

.dot:nth-child(3) {
  width: 160px;
  height: 160px;
  margin-top: -80px;
  margin-left: -80px;
  border-width: 20px;
  border-color: #6bb2cd;
  animation-name: dot3;
  animation-delay: 150ms;
  box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}
#searchInput{
  height: 40px;
  margin-left: 20px;
  margin-bottom: 20px;
  border-radius: 12px;
}
#searchButton {
  position: relative;
  width: 100px;

  display: inline-block;
  padding: 10px 20px;
  font-size: 15px;
  font-weight: bold;
  text-align: center;
  text-decoration: none;
  color: #fff;
  background-color: #ff5252;
  border: 2px solid #000;
  border-radius: 10px;
  box-shadow: 5px 5px 0px #000;
  transition: all 0.3s ease;
}

#searchButton:hover {
```



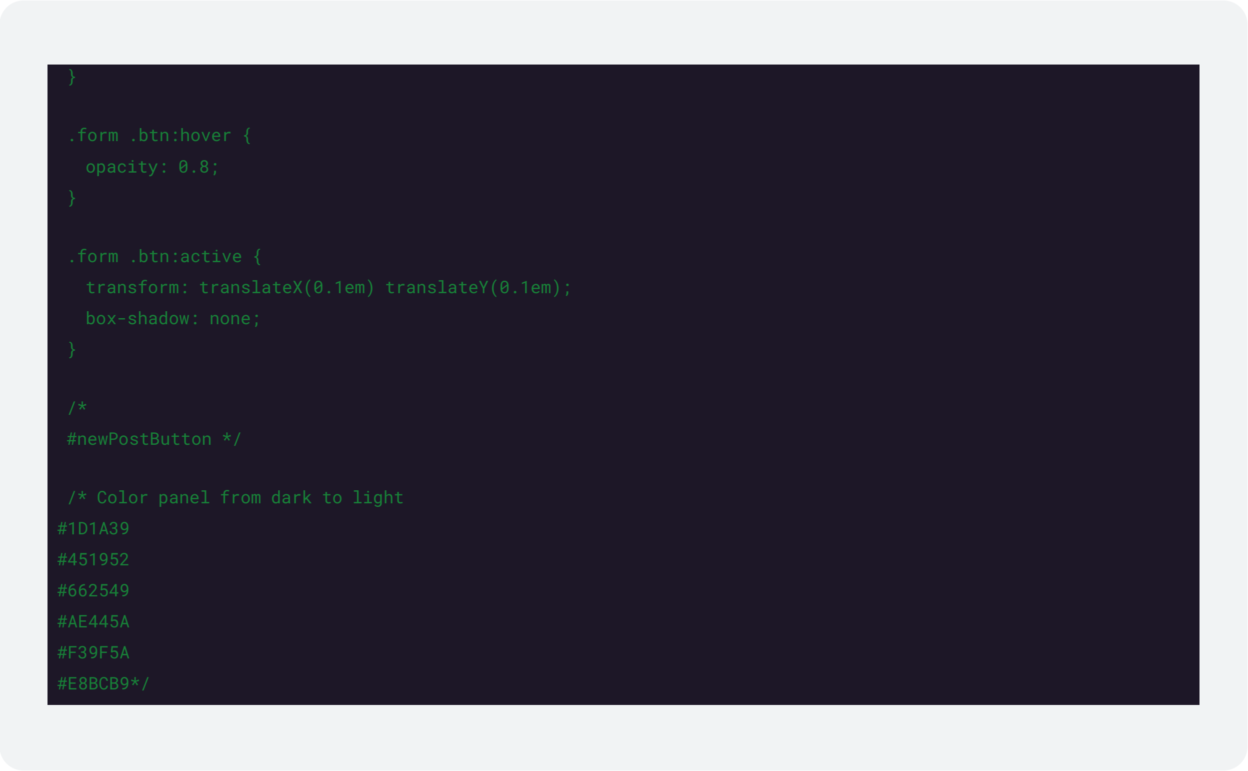
```
  background-color: #fff;
  color: #ff5252;
  border: 2px solid #ff5252;
  box-shadow: 5px 5px 0px #ff5252;
}

#searchButton:active {
  background-color: #fcf414;
  box-shadow: none;
  transform: translateY(4px);
}
#searchButton a{
  text-decoration: none;
  color: white;
}
#searchButton a:hover{
  color: #ff5252;
}
.loader {
    position: fixed;
    top: 0;
    left: 0;
    width: 100vw;
    height: 100vh;
    display: flex;
    align-items: center;
    justify-content: center;
    background: #333333;
    transition: opacity 2s, visibility 2s;
  }

  .loader--hidden {
    opacity: 0;
    visibility: hidden;
  }



  @keyframes loading {
    from {
      transform: rotate(0turn);
    }
    to {
      transform: rotate(1turn);
    }
  }

@keyframes dot1 {
  0% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
```



```
  50% {
    border-color: #1e574f;
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot2 {
  0% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #449180;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot3 {
  0% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #6bcdb2;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

.commentText{
  margin-left: 22%;
```



```css
  width: 56% ;
  height: 100px;
  margin-bottom: 10px !important;
  border-radius: 5px;
  text-align: center;
}
.commentButton {
  position: relative;
  width: 200px;
  margin-top: 50px;
  margin-left: 40%;
  display: inline-block;
  padding: 10px 20px;
  font-size: 15px;
  font-weight: bold;
  text-align: center;
  text-decoration: none;
  color: #fff;
  background-color: #ff5252;
  border: 2px solid #000;
  border-radius: 10px;
  box-shadow: 5px 5px 0px #000;
  transition: all 0.3s ease;
}

.commentButton:hover {
  background-color: #fff;
  color: #ff5252;
  border: 2px solid #ff5252;
  box-shadow: 5px 5px 0px #ff5252;
}

.commentButton:active {
  background-color: #fcf414;
  box-shadow: none;
  transform: translateY(4px);
}
.commentButton a{
  text-decoration: none;
  color: white;
}
.commentButton a:hover{
  color: #ff5252;
}
/* you can choose any color you want. don't be afraid to experiment  */

.form {
    display: flex;
    flex-direction: column;
    gap: 10px;
    padding-top: 3em;
    padding-left: 2.8em;
  padding-right: 2.8em;
  padding-bottom: 2.1em;
  border: 2px dashed #1e3f57;
  border-radius: 15px;
  background-color: #EADDCA;
  box-shadow: 0 0 0 4px #EADDCA, 2px 2px 4px 2px rgba(0, 0, 0, 0.5);
  transition: .4s ease-in-out;
}

::placeholder {
  color: #1e3f57;
  text-align: center;
}

.form .heading {
  padding-left: 0.8em;
  color: #1e3f57;
  background-color: transparent;
  letter-spacing: .5em;
  text-align: center;
  padding-top: 1em;
  padding-bottom: 1em;
  font-size: 2em;
  text-shadow: inset -1px -1px 1px #1e3f57;
}

.form .input {
  outline: none;
  padding: 0.5em;
  border: 1px solid #1e3f57;
  color: #1e3f57;
  width: 14em;
  height: 3em;
  border-radius: 10px;
  background-color: #EADDCA;
  text-align: center;
}

.form .btn {
  align-self: center;
  margin-top: 2em;
  border-radius: 10px;
  outline: none;
  border: none;
  margin-left: 50%;
  color: white;
  background-color: #E5AA70;
  font-weight: bold;
  letter-spacing: 0.1em;
  transition: .4s ease-in-out opacity,.1s ease-in-out active;
  padding: 1em;
  box-shadow: 0.5px 0.5px 0.5px 0.5px rgba(0, 0, 0, 0.5);
```

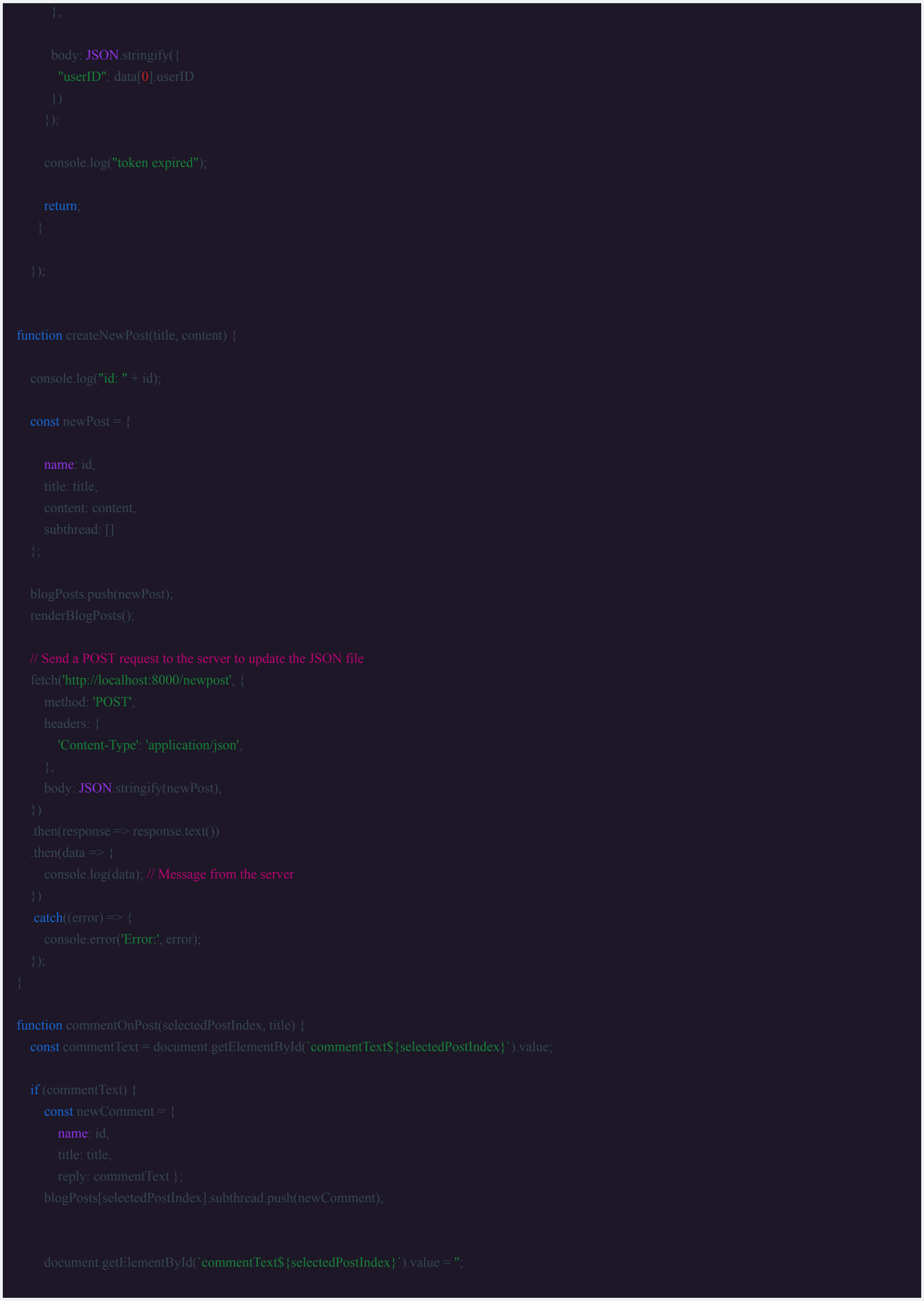


```
 }

 .form .btn:hover {
   opacity: 0.8;
 }

 .form .btn:active {
   transform: translateX(0.1em) translateY(0.1em);
   box-shadow: none;
 }

 /*
 #newPostButton */

 /* Color panel from dark to light
#1D1A39
#451952
#662549
#AE445A
#F39F5A
#E8BCB9*/
```

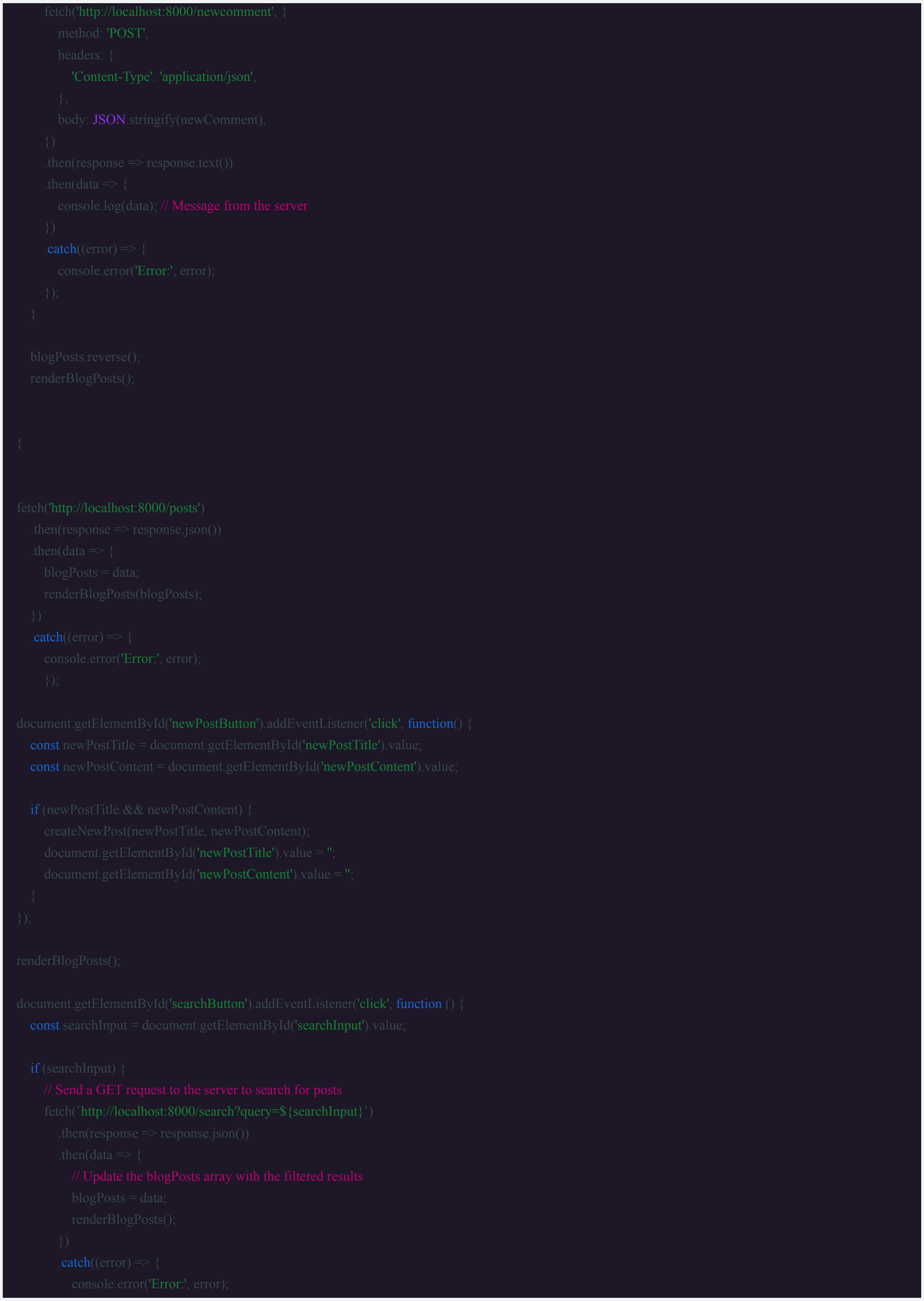
Blog.js

JavaScript

```
document.addEventListener('DOMContentLoaded', function() {
  window.addEventListener("load", () => {
    const loader = document.querySelector(".loader");

    loader.classList.add("loader--hidden");

    loader.addEventListener("transitionend", () => {
     document.body.removeChild(loader);
    });
    });
  let blogPosts = []
  let activePostIndex = null;

  function renderBlogPosts() {
    posts = {};
    const blogPostsContainer = document.getElementById('blogPosts');
    blogPostsContainer.innerHTML = '';
    blogPosts.reverse();

    blogPosts.forEach((post, index) => {
      posts[index] = post.title;
      const postElement = document.createElement('div');
      postElement.classList.add('post');
      postElement.innerHTML = `
        <h2>${post.title} </h2>
        <p>by ${post.name}</p>
        <p>${post.content}</p>
        <button class="toggleReplies" data-index="${index}">Show Replies</button>
        <div class="replies" style="display:none;"></div>
        <input type="text" class="commentText" id="commentText${index}" placeholder="Your Comment" style="display:none;">
        <button class="commentButton" data-index="${index}" style="display:none;">Comment</button>
      `;
      blogPostsContainer.appendChild(postElement);

      const toggleButton = postElement.querySelector('.toggleReplies');
      const repliesContainer = postElement.querySelector('.replies');
      const commentTextInput = postElement.querySelector('.commentText');
      const commentButton = postElement.querySelector('.commentButton');

      toggleButton.addEventListener('click', () => {
        repliesContainer.innerHTML = '';
        post.subthread.forEach(reply => {
          const replyElement = document.createElement('div');
          replyElement.classList.add('reply');

          // Create elements for name and reply
          const nameElement = document.createElement('p');
          nameElement.textContent = `Name: ${reply.name}`;
          nameElement.style.fontSize = '12px'; // Set font size for name

          const replyContent = document.createElement('p');
          replyContent.textContent = `Reply: ${reply.reply}`;
          replyContent.style.fontSize = '12px'; // Set font size for reply

          // Apply border style to create separation between replies
        replyElement.style.border = '1px solid #ccc'; // Adding a border

        // Append name and reply to the replyElement
        replyElement.appendChild(nameElement);
        replyElement.appendChild(replyContent);

        // Append the replyElement to the repliesContainer
        repliesContainer.appendChild(replyElement);
      });

      repliesContainer.style.display = repliesContainer.style.display === 'none' ? 'block' : 'none';

      if (activePostIndex !== null && activePostIndex !== index) {
        const prevCommentTextInput = document.getElementById(`commentText${activePostIndex}`);
        const prevCommentButton = document.querySelector(`.commentButton[data-index="${activePostIndex}"]`);
        prevCommentTextInput.style.display = 'none';
        prevCommentButton.style.display = 'none';
      }

      commentTextInput.style.display = commentTextInput.style.display === 'none' ? 'block' : 'none';
      commentButton.style.display = commentButton.style.display === 'none' ? 'block' : 'none';

      activePostIndex = index;
    });

    commentButton.addEventListener('click', () => {
      commentOnPost(index, posts[index]);
    });
  });
}



var id;

  fetch('http://localhost:8000/tokenGet')
  .then(response => response.json())
  .then(data => {
   if (data.length == 0) {
    console.log("no token");
    // add whatever

    return;
   }
   id = data[0].userID;
   var expire = data[0].expire;
   let now = new Date(); // Get the current date and time
   let hours = now.getHours(); // Get the hours component
   let minutes = now.getMinutes(); // Get the minutes component

   let nowInMinutes = hours * 60 + minutes; // Convert hours and minutes to total minutes

   // console.log("expire: " + expire);
   // console.log("nowInMinutes: " + nowInMinutes);
   if (nowInMinutes > expire) {

    fetch('http://localhost:8000/expiredToken', {
     method: 'post',
     headers: {
      'Content-Type': 'application/json',
```



```
          },

          body: JSON.stringify({
            "userID": data[0].userID
          })
        });

        console.log("token expired");

        return;
      }

    });


function createNewPost(title, content) {

    console.log("id: " + id);

    const newPost = {

        name: id,
        title: title,
        content: content,
        subthread: []
    };

    blogPosts.push(newPost);
    renderBlogPosts();

    // Send a POST request to the server to update the JSON file
    fetch('http://localhost:8000/newpost', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify(newPost),
    })
    .then(response => response.text())
    .then(data => {
        console.log(data); // Message from the server
    })
    .catch((error) => {
        console.error('Error:', error);
    });
}

function commentOnPost(selectedPostIndex, title) {
    const commentText = document.getElementById(`commentText${selectedPostIndex}`).value;

    if (commentText) {
        const newComment = {
            name: id,
            title: title,
            reply: commentText };
        blogPosts[selectedPostIndex].subthread.push(newComment);



        document.getElementById(`commentText${selectedPostIndex}`).value = '';
```



```
        fetch('http://localhost:8000/newcomment', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(newComment),
        })
        .then(response => response.text())
        .then(data => {
            console.log(data); // Message from the server
        })
        .catch((error) => {
            console.error('Error:', error);
        });
    }

    blogPosts.reverse();
    renderBlogPosts();


}



fetch('http://localhost:8000/posts')
    .then(response => response.json())
    .then(data => {
        blogPosts = data;
        renderBlogPosts(blogPosts);
    })
    .catch((error) => {
        console.error('Error:', error);
        });

document.getElementById('newPostButton').addEventListener('click', function() {
    const newPostTitle = document.getElementById('newPostTitle').value;
    const newPostContent = document.getElementById('newPostContent').value;

    if (newPostTitle && newPostContent) {
        createNewPost(newPostTitle, newPostContent);
        document.getElementById('newPostTitle').value = '';
        document.getElementById('newPostContent').value = '';
    }
});

renderBlogPosts();

document.getElementById('searchButton').addEventListener('click', function () {
    const searchInput = document.getElementById('searchInput').value;

    if (searchInput) {
        // Send a GET request to the server to search for posts
        fetch(`http://localhost:8000/search?query=${searchInput}`)
            .then(response => response.json())
            .then(data => {
                // Update the blogPosts array with the filtered results
                blogPosts = data;
                renderBlogPosts();
            })
            .catch((error) => {
                console.error('Error:', error);
        });
    } else {
      // If search input is empty, reload all blog posts
      fetch('http://localhost:8000/posts')
        .then(response => response.json())
        .then(data => {
          blogPosts = data;
          renderBlogPosts();
        })
        .catch((error) => {
          console.error('Error:', error);
        });
    }
  });

});
```

Online.html

Unset

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/x-icon" href="resources/se.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="/static/online.css" />

    <title>Online Resources</title>
  </head>

  <body>
    <div class="container">
      <div class="main1">
        <div class="side">
          <h1>Online Resources</h1>
          <div class="side-content">
            <p>
              <a class="odd" href="https://www.w3schools.com/">W3Schools</a>
            </p>
            <p>
              <a href="https://www.codecademy.com/">Codecademy</a>
            </p>
            <p>
              <a class="odd" href="https://www.freecodecamp.org/"
                >FreeCodeCamp</a
              >
            </p>
            <p>
              <a href="https://www.udemy.com/">Udemy</a>
            </p>
            <p>
              <a class="odd" href="https://www.khanacademy.org/"
                >Khan Academy</a
        >
      </p>
      <p>
        <a href="https://www.codeavengers.com/">Code Avengers</a>
      </p>
      <p>
        <a class="odd" href="https://www.code.org/">Code.org</a>
      </p>
      <p>
        <a href="https://www.codecombat.com/">CodeCombat</a>
      </p>
      <p>
        <a class="odd" href="https://www.codeconquest.com/"
          >Code Conquest</a
        >
      </p>
      <p>
        <a href="https://www.codeasy.net/">Codeasy.net</a>
      </p>
    </div>
  </div>

  <button class="comic-button3">
    <a href="./se.html"> Back</a>
  </button>
</div>
<div class="main2">
  <div class="body1">
    <div class="head-video">
      <input
        type="text"
        placeholder="Search.."
        class="search"
        id="searchInput"
      />
      <button class="comic-button1" id="searchButton">Search</button>
      <form id="addVideoForm">
```

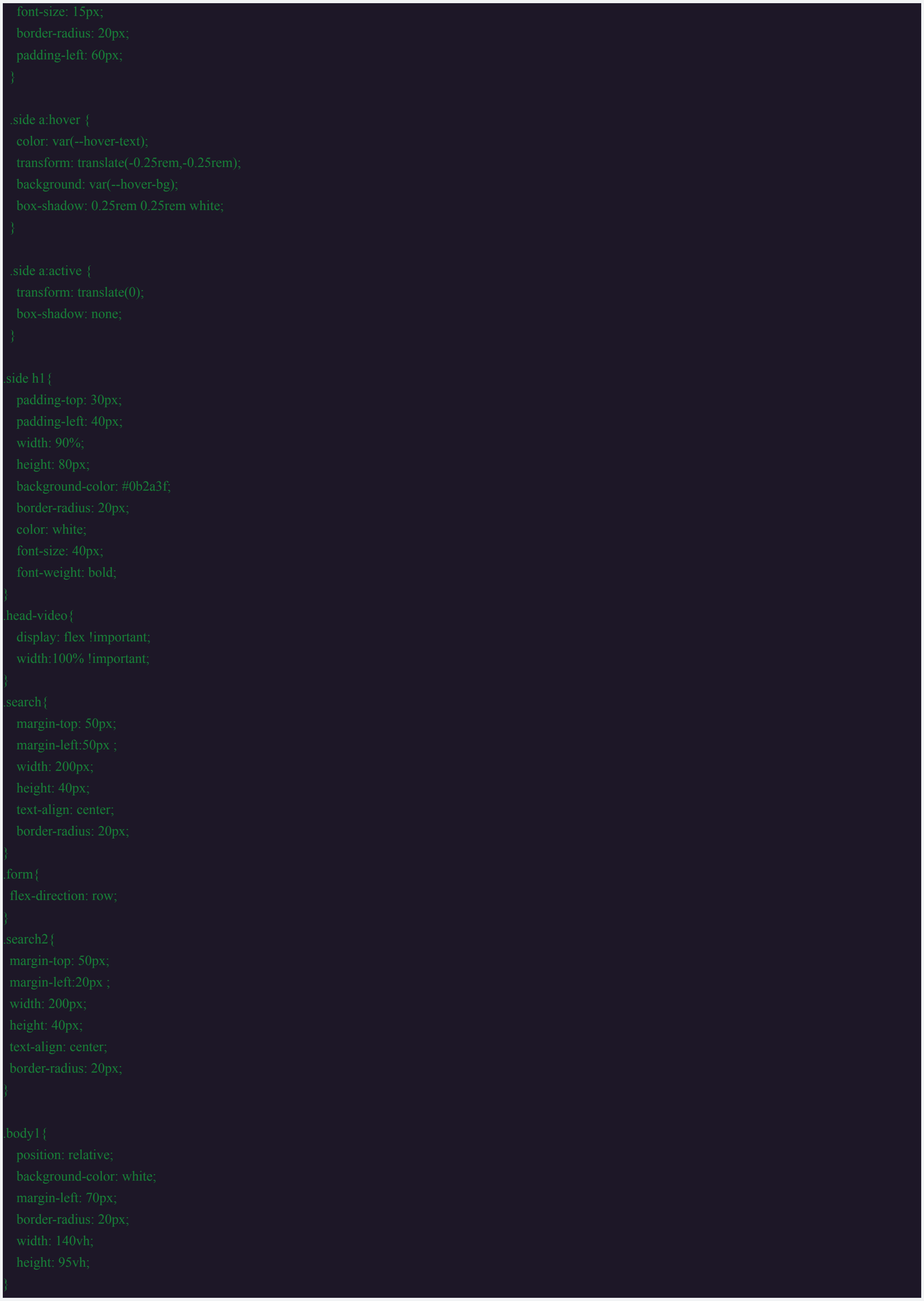
```
       <input
        class="search2"
        type="text"
        id="videoUrl"
        placeholder="Video URL"
       />
       <input
        class="search2"
        type="text"
        id="videoTitle"
        placeholder="Video Title"
       />
       <button class="comic-button1" type="submit" id="submit">
        Add Video
       </button>
      </form>
      <button class="comic-button2">
       <a href="./calendar.html"> Set Schedule</a>!
      </button>
     </div>
     <div class="inner-video" id="videosContainer">
      <div class="video1">
       <iframe
        src="https://www.youtube.com/embed/xX0iomJ6I8U?si=owhzrKMcY8WZ8TZI"
        title="YouTube video player"
        frameborder="0"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
        allowfullscreen
       ></iframe>
      </div>
     </div>
    </div>
   </div>
  </div>
 </body>
 <script src="/static/online.js"></script>
```

```
</html>
```

Online.css

Unset

```
.container{
    width: 100%;
    height: 100%;
    background-size: cover;

    display: flex;
    flex-direction: row;
    background-color: white;


}
h2{
  color: #1D1A39 !important;
  padding-left: 20px;
  font-size: 20px;
  border-top-right-radius: 40px;
  background-color: #fff;
  width:300px;
}
.main1{
    direction: flex;
}
.side{
    border-radius: 20px;

    margin-top: 10px;
    background-color: #1e3f57;
    position: fixed;
    position: relative;
    width: 150%;
    height: 85vh;

    filter: brightness(1.3);
    font-family: 'Roboto', sans-serif;


}

.side a{

    color: white;
    padding: 10px;

    text-decoration: none;
    font-size: 20px;
    width: 40%;
    font-weight: bold;
    display: block;
}

.side a {
    --hover-bg: #306d99;
    --hover-text: #ffffff;
    color: #fff;
    border: 1px solid var(--bg);
    border-radius: 4px;
    background: var(--bg);
    transition: 0.2s;
```



```
  font-size: 15px;
  border-radius: 20px;
  padding-left: 60px;
 }

 .side a:hover {
  color: var(--hover-text);
  transform: translate(-0.25rem,-0.25rem);
  background: var(--hover-bg);
  box-shadow: 0.25rem 0.25rem white;
 }

 .side a:active {
  transform: translate(0);
  box-shadow: none;
 }

.side h1{
  padding-top: 30px;
  padding-left: 40px;
  width: 90%;
  height: 80px;
  background-color: #0b2a3f;
  border-radius: 20px;
  color: white;
  font-size: 40px;
  font-weight: bold;
}
.head-video{
  display: flex !important;
  width:100% !important;
}
.search{
  margin-top: 50px;
  margin-left:50px ;
  width: 200px;
  height: 40px;
  text-align: center;
  border-radius: 20px;
}
.form{
 flex-direction: row;
}
.search2{
 margin-top: 50px;
 margin-left:20px ;
 width: 200px;
 height: 40px;
 text-align: center;
 border-radius: 20px;
}

.body1{
  position: relative;
  background-color: white;
  margin-left: 70px;
  border-radius: 20px;
  width: 140vh;
  height: 95vh;
}
```

```css
.loader1 {
 margin-top: 100px;
 width: 100%;
 height: auto;
 perspective: 200px;
 z-index:9999 !important;
 position: absolute;

}



/* The dot */
.dot {
 position: absolute;
 top: 50%;
 left: 50%;
 width: 120px;
 height: 120px;
 margin-top: -60px;
 margin-left: -60px;
 border-radius: 100px;
 border: 40px outset #1e3f57;
 transform-origin: 50% 50%;
 transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
 background-color: transparent;
 animation: dot1 1000ms cubic-bezier(.49,.06,.43,.85) infinite;
}

.dot:nth-child(2) {
 width: 140px;
 height: 140px;
 margin-top: -70px;
 margin-left: -70px;
 border-width: 30px;
 border-color: #447891;
 animation-name: dot2;
 animation-delay: 75ms;
 box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
 transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}

.dot:nth-child(3) {
 width: 160px;
 height: 160px;
 margin-top: -80px;
 margin-left: -80px;
 border-width: 20px;
 border-color: #6bb2cd;
 animation-name: dot3;
 animation-delay: 150ms;
 box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
 transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}
.loader {
   position: fixed;
   top: 0;
   left: 0;
   width: 100vw;
   height: 100vh;
   display: flex;
```

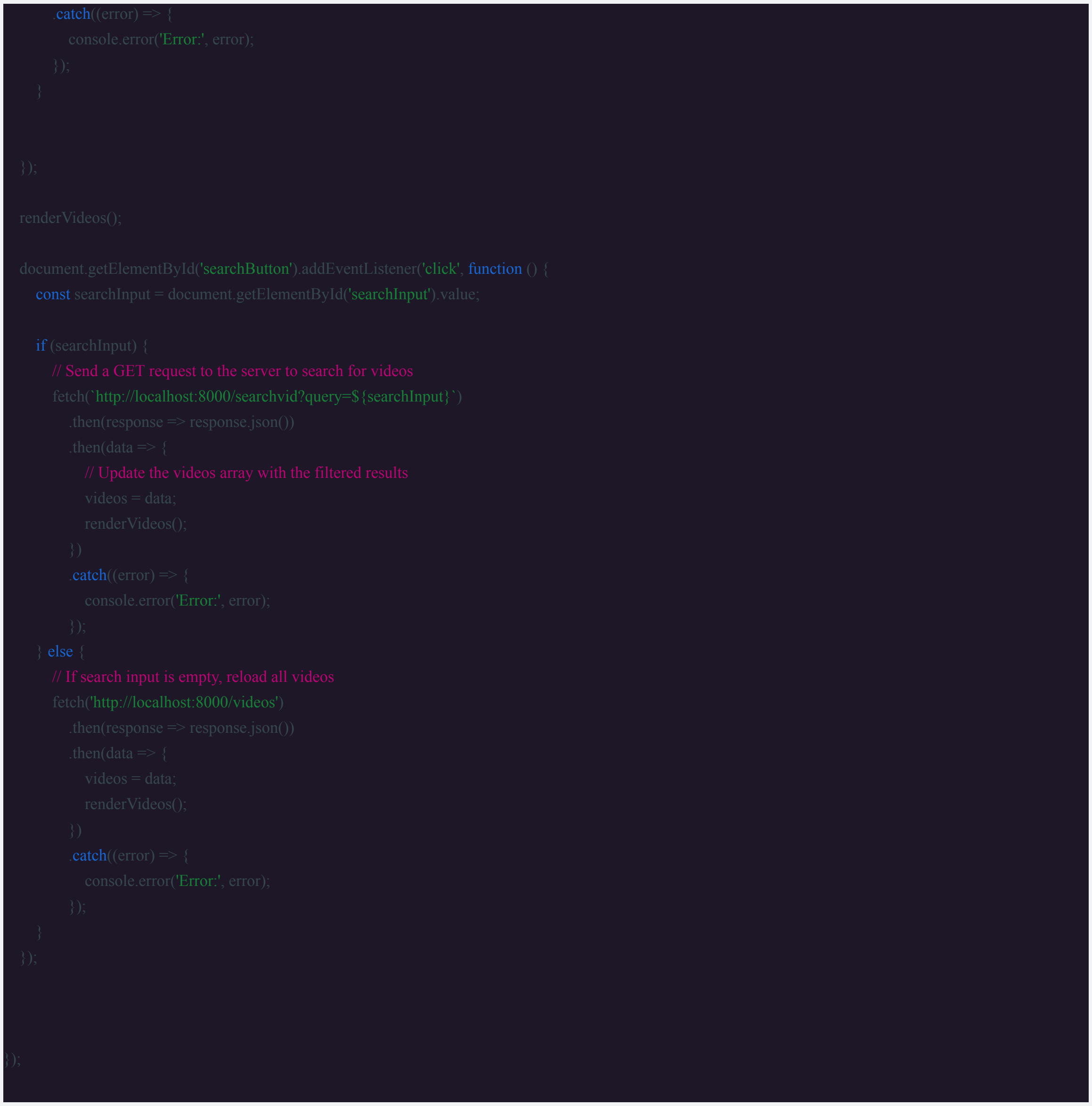
```
    align-items: center;
    justify-content: center;
    background: #333333;
    transition: opacity 2s, visibility 2s;
  }

  .loader--hidden {
    opacity: 0;
    visibility: hidden;
  }



  @keyframes loading {
    from {
      transform: rotate(0turn);
    }
    to {
      transform: rotate(1turn);
    }
  }

@keyframes dot1 {
  0% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #1e574f;
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot2 {
  0% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #449180;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot3 {
  0% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #6bcdb2;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}
.comic-button2 {
    width: 100px;
    height: 60px;
    margin-top: 30px;
    margin-left: 20px;
    display: inline-block;
    padding: 10px 20px;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
  }

  .comic-button2:hover {
    background-color: #fff;
    color: #ff5252;
    border: 2px solid #ff5252;
    box-shadow: 5px 5px 0px #ff5252;
  }

  .comic-button2:active {
    background-color: #fcf414;
    box-shadow: none;
    transform: translateY(4px);
  }
  .comic-button2 a{
    text-decoration: none;
    color: white;
  }
  .comic-button2 a:hover{
    color: #ff5252;
  }
  .odd{
    margin-left: 120px !important;
  }
  input[type="file"] {
    display: none;
  }
  label{
    width: 150px;
    margin-top: 50px;
    margin-left: 300px;
    display: inline-block;
    padding: 10px 20px;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
  }

  label:hover {
    background-color: #fff;
    color: #ff5252;
    border: 2px solid #ff5252;
    box-shadow: 5px 5px 0px #ff5252;
  }

  label:active {
    background-color: #fcf414;
    box-shadow: none;
    transform: translateY(4px);
  }
  label  a{
    text-decoration: none;
    color: white;
  }
  label a:hover{
    color: #ff5252;
  }
.comic-button1 {
    display: inline-block;
    width: 100px;
    height: 50px;
    margin-top: 40px;
    margin-left: 10px;
    margin-right: 10px;
    padding: 10px 20px;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
  }

  .comic-button1:hover {
    background-color: #fff;
    color: #ff5252;
    border: 2px solid #ff5252;
    box-shadow: 5px 5px 0px #ff5252;
  }

  .comic-button1:active {
    background-color: #fcf414;
    box-shadow: none;
    transform: translateY(4px);
  }
  .comic-button1 a{
    text-decoration: none;
    color: white;
  }
  .comic-button1 a:hover{
    color: #ff5252;
  }

  .comic-button3 {
    width: 100px;
    height: 40px;
    margin-left: 30%;;
    display: inline-block;
    font-size: 15px;
    font-weight: bold;
    text-align: center;
    text-decoration: none;
    color: #fff;
    background-color: #ff5252;
    border: 2px solid #000;
    border-radius: 10px;
    box-shadow: 5px 5px 0px #000;
    transition: all 0.3s ease;
    z-index: 3000 !important;
  }

  .comic-button3:hover {
    background-color: #fff !important;
    color: #ff5252 !important;
    border: 2px solid #ff5252 !important;
    box-shadow: 5px 5px 0px #ff5252 !important;
  }

  .comic-button3:active {
    background-color: #fcf414 !important;
    box-shadow: none !important ;
    transform: translateY(4px) !important ;
  }
  .comic-button3 a{
    text-decoration: none !important ;
    color: white !important;
  }
  .comic-button3 a:hover{
    color: #ff5252 !important;
  }
  .inner-video{
    margin-top: 50px;
    margin-left: 40px;
    width: 900px;
```

```
    height: 470px;
    border-radius: 20px;
    border: #1D1A39 2px solid;
    background-color: transparent;
    display: flex;
    flex-direction: column;
    padding: 50px;
    overflow-y: auto;
  }
  .video1{
    margin-top: 30px;
    margin-left: 33px;
    width: 420px;
    height: 310px;
    border-radius: 20px;
    border: #1D1A39 2px solid;
    display: flex;
    justify-content: center;
    background-color: white;
  }

.video1 iframe{
    padding-top: 5px;
    border-radius: 20px;
    width: 390px;
    height: 300px;
}
.video{
  margin-left: 20%;
}

  .body1{
    border: #1D1A39 2px solid;
  }

  .inner-video::-webkit-scrollbar {
    width: 12px; /* width of the scrollbar */
  }

  .inner-video::-webkit-scrollbar-thumb {
    background-color: #1D1A39; /* color of the thumb */
    border-radius: 10px; /* roundness of the thumb */
  }

  .inner-video::-webkit-scrollbar-track {
    background-color: transparent; /* color of the track */
  }
/* #1D1A39
 #451952
 #662549
 #AE445A
 #F39F5A
 #E8BCB9 */
```

Online.js

JavaScript

```javascript
let videos;

document.addEventListener('DOMContentLoaded', function() {

  fetch('http://localhost:8000/videos')
    .then(response => response.json())
    .then(data => {
      videos = data;
      renderVideos();
    })
    .catch((error) => {
      console.error('Error:', error);
    });
  // Function to render videos
  function renderVideos() {
    const videosContainer = document.getElementById('videosContainer');
    videosContainer.innerHTML = '';

    if (videos) {
      videos.forEach((video, index) => {
        const videoElement = document.createElement('div');
        videoElement.classList.add('video');
        videoElement.innerHTML = `
          <h2>${video.title}</h2>
          ${video.url}`;
        videosContainer.appendChild(videoElement);
      });
    } else {
      // Handle the case where videos is undefined (perhaps show a message or take other actions)
      console.error('Videos data is undefined.');
    }
  }

  // Function to handle form submission
  document.getElementById('submit').addEventListener('click', function(event) {
    event.preventDefault();
    const videoUrl = document.getElementById('videoUrl').value;
    const videoTitle = document.getElementById('videoTitle').value;

    if (videoUrl && videoTitle) {
      let videopost = { url: videoUrl, title: videoTitle };
      videos.push(videopost);
      renderVideos();
      document.getElementById('videoUrl').value = '';
      document.getElementById('videoTitle').value = '';
      fetch('http://localhost:8000/add_video', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(videopost),
      })
      .then(response => response.text())
      .then(data => {
        console.log(data); // Message from the server
      })
```



```
            .catch((error) => {
                console.error('Error:', error);
            });
        }


    });

    renderVideos();

    document.getElementById('searchButton').addEventListener('click', function () {
        const searchInput = document.getElementById('searchInput').value;

        if (searchInput) {
            // Send a GET request to the server to search for videos
            fetch(`http://localhost:8000/searchvid?query=${searchInput}`)
                .then(response => response.json())
                .then(data => {
                    // Update the videos array with the filtered results
                    videos = data;
                    renderVideos();
                })
                .catch((error) => {
                    console.error('Error:', error);
                });
        } else {
            // If search input is empty, reload all videos
            fetch('http://localhost:8000/videos')
                .then(response => response.json())
                .then(data => {
                    videos = data;
                    renderVideos();
                })
                .catch((error) => {
                    console.error('Error:', error);
                });
        }
    });



});
```

Calendar.html

Unset

```
<!DOCTYPE html>
<html>
  <head>
    <title>calendar</title>
    <link rel="stylesheet" href="/static/calendar.css" />
    <script src="/static/calendar.js"></script>
  </head>
  <body>
    <div class="loader">
      <div class="loader1">
        <div class="dot"></div>
        <div class="dot"></div>
        <div class="dot"></div>
      </div>
    </div>
    <div class="calendar1" id="calendarContainer">
      <div class="container1">
        <div class="left">
          <div class="calendar">
            <div class="month">
              <i class="fas fa-angle-left prev"></i>
              <div class="date">december 2015</div>
              <i class="fas fa-angle-right next"></i>
            </div>
            <div class="weekdays">
              <div>Sun</div>
              <div>Mon</div>
              <div>Tue</div>
              <div>Wed</div>
              <div>Thu</div>
              <div>Fri</div>
              <div>Sat</div>
            </div>
            <div class="days"></div>
            <div class="goto-today">
              <div class="goto">
                <input
                  type="text"
                  placeholder="mm/yyyy"
                  class="date-input"
                  width="60px"
                />
                <button class="goto-btn">Go</button>
              </div>
              <button class="back-btn">
                <a href="./online.html">Back</a>
              </button>
              <button class="today-btn">Today</button>
            </div>
          </div>
        </div>
        <div class="right">
          <div class="today-date">
            <div class="event-day"></div>
            <div class="event-date"></div>
          </div>
```

```
     <div class="events"></div>
     <div class="add-event-wrapper">
      <div class="add-event-header">
       <div class="title">Add Event</div>
       <i class="fas fa-times close"></i>
      </div>
      <div class="add-event-body">
       <div class="add-event-input">
        <input
         type="text"
         placeholder="Event Name"
         class="event-name"
        />
       </div>
       <div class="add-event-input">
        <input
         type="text"
         placeholder="Event Time From"
         class="event-time-from"
        />
       </div>
       <div class="add-event-input">
        <input
         type="text"
         placeholder="Event Time To"
         class="event-time-to"
        />
       </div>
      </div>
      <div class="add-event-footer">
       <button class="add-event-btn">Add Event</button>
      </div>
     </div>
    </div>
    <button class="add-event">
     +
     <i class="fas fa-plus"></i>
    </button>
   </div>
  </div>
 </body>
</html>
```

Calendar.css

Unset

```
/* ::-webkit-scrollbar {
  width: 5px;
 }
 ::-webkit-scrollbar-track {
  background: #f5f5f5;
  border-radius: 50px;
 }
```

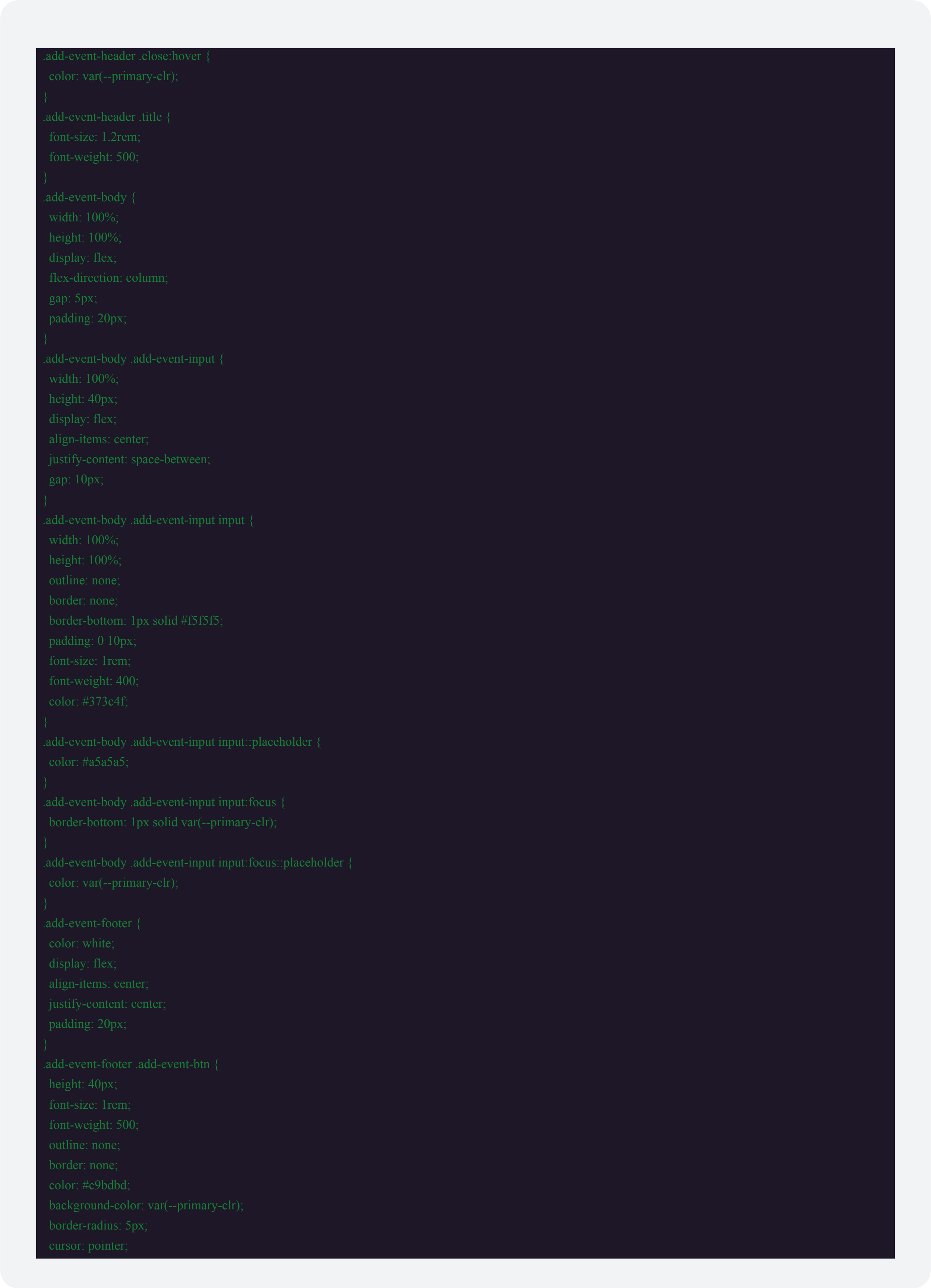
```
.scrollbar-thumb {
  background: var(--primary-clr);
  border-radius: 50px;
} */

body{
  position: relative;
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  padding-bottom: 30px;
  background-color: #e2e1dc;
}
.container1 {
  position: relative;
  width: 1400px;
  min-height: 850px;
  margin: 0 auto;
  padding: 5px;
  color: #fff;
  display: flex;

  border-radius: 10px;
  background-color: #1e3f57;
}
.left {
  width: 60%;
  padding: 20px;
}
.calendar {
  position: relative;
  width: 105%;
  height: 100%;
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  justify-content: space-between;
  color: #1D1A39;
  border-radius: 5px;
  background-color: #7590a4;
}
/* set after behind the main element */
.calendar::before,
.calendar::after {
  content: "";
  position: absolute;
  top: 50%;
  left: 100%;
  width: 12px;
  height: 97%;
  border-radius: 0 5px 5px 0;
  background-color: #d3d4d6d7;
  transform: translateY(-50%);
}
.calendar::before {
  height: 94%;
  left: calc(100% + 12px);
  background-color: rgb(153, 153, 153);
}
```

```
.calendar .month {
  width: 100%;
  height: 150px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  font-size: 2rem;
  font-weight: bold;
  text-transform: capitalize;
}
.calendar .month .prev,
.calendar .month .next {
  cursor: pointer;
}
.calendar .month .prev:hover,
.calendar .month .next:hover {
  color: var(--primary-clr);
}
.calendar .weekdays {
  width: 100%;
  height: 100px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
  font-size: 1rem;
  font-weight: 500;
  text-transform: capitalize;
}
.date{
  color:white;
}
.weekdays div {
  width: 14.28%;
  height: 100%;
  color: white;
  font-size: 1rem;
  display: flex;
  align-items: center;
  justify-content: center;
}
.calendar .days {
  width: 100%;
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  font-size: 1rem;
  font-weight: 500;
  margin-bottom: 20px;
}
.calendar .days .day {
  width: 14%;
  margin: 10px;
  height: 80px;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  color: var(--primary-clr);
  border-radius: 10px;
```

```
}
/* .calendar .days .day:nth-child(7n + 1) {
  border-left: 2px solid #1D1A39;
}
.calendar .days .day:nth-child(7n) {
  border-right: 2px solid #1D1A39;
}
.calendar .days .day:nth-child(-n + 7) {
  border-top: 2px solid #1D1A39;
}
.calendar .days .day:nth-child(n + 29) {
  border-bottom: 2px solid #1D1A39;
}
 */
.calendar .days .day:not(.prev-date, .next-date):hover {
  color: #fff;
  background-color: var(--primary-clr);
}
.calendar .days .prev-date,
.calendar .days .next-date {
  color: #b3b3b3;
}
.calendar .days .active {
  position: relative;
  font-size: 2rem;
  color: white;
  background-color: #ff5252;
}
.calendar .days .active::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  box-shadow: 0 0 10px 2px var(--primary-clr);
}
.calendar .days .today {
  font-size: 2rem;
}
.calendar .days .event {
  position: relative;
}
.calendar .days .event::after {
  content: "";
  position: absolute;
  bottom: 10%;
  left: 50%;
  width: 75%;
  height: 6px;
  border-radius: 10px;
  transform: translateX(-50%);
  background-color: var(--primary-clr);
}
.calendar .days .day:hover.event::after {
  background-color: #fff;
}
.calendar .days .active.event::after {
  background-color: #fff;
  bottom: 20%;
```

```
}
.calendar .days .active.event {
  padding-bottom: 10px;
}
.calendar .goto-today {
  width: 90%;
  height: 50px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 5px;
  padding: 0 20px;
  margin-bottom: 20px;
  color: var(--primary-clr);
}
.calendar .goto-today .goto {
  display: flex;
  align-items: center;
  border-radius: 5px;
  overflow: hidden;
  border: 1px solid var(--primary-clr);
}
.today-btn {
  color: purple !important;
  text-transform: uppercase !important;
  text-decoration: none !important;
  border: 2px solid purple !important;
  padding: 10px 20px !important;
  font-size: 17px !important ;
  font-weight: bold !important;
  position: relative !important;
  transition: all 1s !important;
  overflow: hidden !important;
}
.day{
    font-weight: 700;
    background: rgb(255, 255, 255);
    color: blueviolet;
    border-radius: .5rem;
    border: hidden;
    transition-duration: 1s;
    transition-property: border-top,
       border-left,
       border-bottom,
       border-right,
       box-shadow;
   }
.day:hover{

    border-top: 2px solid blueviolet;
    border-left: 2px solid blueviolet;
    border-bottom: 2px solid rgb(238, 103, 238);
    border-right: 2px solid rgb(238, 103, 238);
    box-shadow: rgba(240, 46, 170, 0.4) 5px 5px, rgba(240, 46, 170, 0.3) 10px 10px, rgba(240, 46, 170, 0.2) 15px 15px;
   }
.today-btn:hover {
  background-color: purple !important;
  color: white !important;
}
.goto-btn {
  color: purple !important;
  text-transform: uppercase !important;
  text-decoration: none !important;
  text-align: center !important;
  border: 2px solid purple !important;
  padding: 15px 30px !important;
  font-size: 17px !important ;
  font-weight: bold !important;
  position: relative !important;
  transition: all 1s !important;
  overflow: hidden !important;
}

.goto-btn:hover {
  background-color: purple !important;
  color: white !important;
}

.goto-btn::before {
  content: " !important;
  position: absolute !important;
  height: 100%  !important;
  width: 0% !important;
  top: 0 ;
  left: -40px;
  transform: skewX(45deg)  !important;
  background-color: purple !important;
  z-index: -1 !important;
  transition: all 1s !important;
}

.goto-btn:hover::before {
  width: 160%;
}
.calendar .goto-today .goto input {
  width: 100%;
  height: 30px;
  outline: none;
  border: none;
  border-radius: 5px;
  padding: 0 20px;
  color: var(--primary-clr);
  border-radius: 5px;
}
.calendar .goto-today button {
  margin-left: 10px;
  border: 1px solid var(--primary-clr);
  border-radius: 5px;
  cursor: pointer;
  color: var(--primary-clr);
}
.calendar .goto-today button:hover {
  color: #fff;
  background-color: var(--primary-clr);
}
.calendar .goto-today .goto button::before {
  content: ";
position: absolute;
height: 100%;
width: 0%;
```

```css
top: 0;
left: -40px;
transform: skewX(45deg);
background-color: purple;
z-index: -1;
transition: all 1s;
}
.calendar .goto-today .goto button {
  border: none;
  border-radius: 10px;
  border-left: 1px solid var(--primary-clr);
}
.container1 .right {
  position: relative;
  width: 40%;
  min-height: 100%;
  padding: 20px 0;
}

.right .today-date {
  width: 80%;
  height: 50px;
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
  align-items: center;
  justify-content: space-between;
  padding: 0 40px;
  padding-left: 70px;
  margin-top: 50px;
  margin-bottom: 20px;
  text-transform: capitalize;
}
.right .today-date .event-day {
  font-size: 2rem;
  font-weight: 500;
}
.right .today-date .event-date {
  font-size: 1rem;
  font-weight: 400;
  color: #f8f8f9;
}
.events {
  width: 100%;
  height: 100%;
  max-height: 600px;
  overflow-x: hidden;
  overflow-y: auto;
  display: flex;
  flex-direction: column;
  padding-left: 4px;
}
.events .event {
  position: relative;
  margin-left: 50px;
  min-height: 70px;
  display: flex;
  justify-content: center;
  flex-direction: column;
  gap: 5px;
  padding: 0 20px;
  color: #fff;
  background: linear-gradient(90deg, #3f4458, transparent);
  cursor: pointer;
}
/* even event */
.events .event:nth-child(even) {
  background: transparent;
}
.events .event:hover {
  background: linear-gradient(90deg, var(--primary-clr), transparent);
}
.events .event .title {
  display: flex;
  align-items: center;
  pointer-events: none;
}
.events .event .title .event-title {
  font-size: 1rem;
  font-weight: 400;
  margin-left: 20px;
}
.events .event i {
  color: white;
  font-size: 0.5rem;
}
.events .event:hover i {
  color: #fff;
}
.events .event .event-time {
  font-size: 0.8rem;
  font-weight: 400;
  color: #1D1A39;
  margin-left: 15px;
  pointer-events: none;
}
.events .event:hover .event-time {
  color: #fff;
}
/* add tick in event after */
.events .event::after {
  content: "Done";
  position: absolute;
  top: 50%;
  right: 0;
  font-size: 2rem;
  line-height: 1;
  display: none;
  align-items: center;
  justify-content: center;
  opacity: 0.3;
  color: var(--primary-clr);
  transform: translateY(-50%);
}
.events .event:hover::after {
  display: flex;
}
.add-event {
  position: absolute;
  bottom: 30px;
  right: 30px;
  width: 40px;
  height: 40px;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 1rem;
  border: 2px solid #ffffff;
  opacity: 0.5;
  border-radius: 50%;
  background-color: transparent;
  cursor: pointer;
  opacity: 1;

}
.add-event:hover {
  opacity: 1;
}
.add-event i {
  pointer-events: none;
}
.events .no-event {
  width: 100%;
  height: 100%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 1.5rem;
  font-weight: 500;
  color:white;
}
.add-event-wrapper {
  position: absolute;
  bottom: 100px;
  left: 50%;
  width: 90%;
  max-height: 0;
  overflow: hidden;
  border-radius: 5px;
  background-color: #fff;
  transform: translateX(-50%);
  transition: max-height 0.5s ease;
}
.add-event-wrapper.active {
  max-height: 300px;
}
.add-event-header {
  width: 100%;
  height: 50px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0 20px;
  color: #373c4f;
  border-bottom: 1px solid #f5f5f5;
}
.add-event-header .close {
  font-size: 1.5rem;
  cursor: pointer;
}
```



```
.add-event-header .close:hover {
  color: var(--primary-clr);
}
.add-event-header .title {
  font-size: 1.2rem;
  font-weight: 500;
}
.add-event-body {
  width: 100%;
  height: 100%;
  display: flex;
  flex-direction: column;
  gap: 5px;
  padding: 20px;
}
.add-event-body .add-event-input {
  width: 100%;
  height: 40px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 10px;
}
.add-event-body .add-event-input input {
  width: 100%;
  height: 100%;
  outline: none;
  border: none;
  border-bottom: 1px solid #f5f5f5;
  padding: 0 10px;
  font-size: 1rem;
  font-weight: 400;
  color: #373c4f;
}
.add-event-body .add-event-input input::placeholder {
  color: #a5a5a5;
}
.add-event-body .add-event-input input:focus {
  border-bottom: 1px solid var(--primary-clr);
}
.add-event-body .add-event-input input:focus::placeholder {
  color: var(--primary-clr);
}
.add-event-footer {
  color: white;
  display: flex;
  align-items: center;
  justify-content: center;
  padding: 20px;
}
.add-event-footer .add-event-btn {
  height: 40px;
  font-size: 1rem;
  font-weight: 500;
  outline: none;
  border: none;
  color: #c9bdbd;
  background-color: var(--primary-clr);
  border-radius: 5px;
  cursor: pointer;
```

```
    padding: 5px 10px;
    border: 1px solid var(--primary-clr);
  }
  .add-event-btn{
    color: white;
    background-color: #ff5252;
  }
  .add-event-footer .add-event-btn:hover {
    color: white !important;
    background-color: #ff5252 !important;
    box-shadow: 10px 10px 0px #1D1A39 !important;
    top: -5px !important;
    left: -5px !important;
  }
  .add-event{
    color: white;
  }
.event{
  width:70%;
}
.back-btn {
  width: 100px;
  height: 40px;
  margin-left: 50px;
  display: inline-block;
  font-size: 15px;
  font-weight: bold;
  text-align: center;
  text-decoration: none;
  color: #fff;
  background-color: #ff5252;
  border: 2px solid #000;
  border-radius: 10px;
  box-shadow: 5px 5px 0px #000;
  transition: all 0.3s ease;
}
.loader1 {
  margin-top: 100px;
  width: 100%;
  height: auto;
  perspective: 200px;
  z-index:9999 !important;
  position: absolute;

}



/* The dot */
.dot {
  position: absolute;
  top: 50%;
  left: 50%;
  width: 120px;
  height: 120px;
  margin-top: -60px;
  margin-left: -60px;
  border-radius: 100px;
  border: 40px outset #1e3f57;
  transform-origin: 50% 50%;
  transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
```

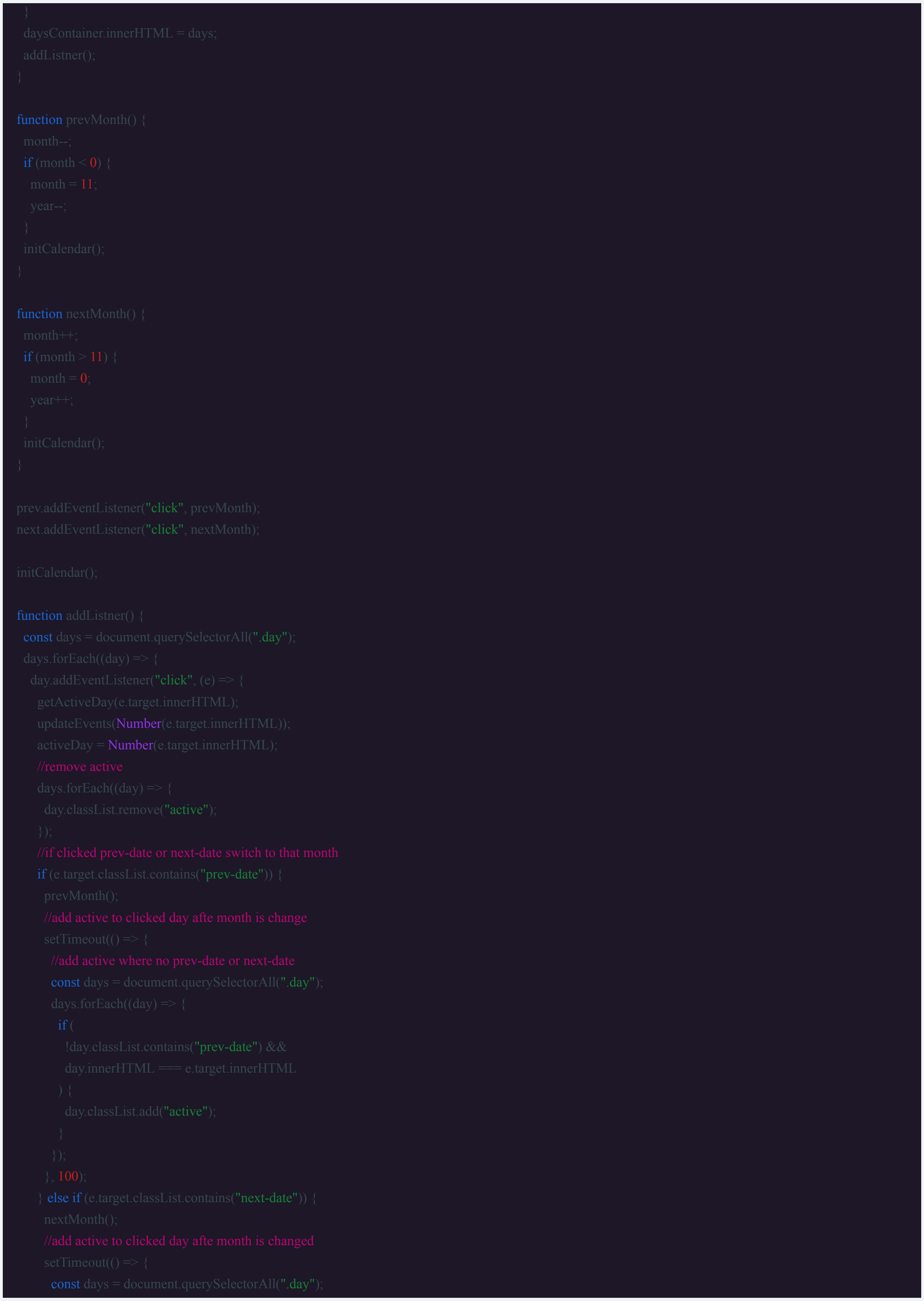
```
 background-color: transparent;
 animation: dot1 1000ms cubic-bezier(.49,.06,.43,.85) infinite;
}

.dot:nth-child(2) {
 width: 140px;
 height: 140px;
 margin-top: -70px;
 margin-left: -70px;
 border-width: 30px;
 border-color: #447891;
 animation-name: dot2;
 animation-delay: 75ms;
 box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
 transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}

.dot:nth-child(3) {
 width: 160px;
 height: 160px;
 margin-top: -80px;
 margin-left: -80px;
 border-width: 20px;
 border-color: #6bb2cd;
 animation-name: dot3;
 animation-delay: 150ms;
 box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
 transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
}
.loader {
  position: fixed;
  top: 0;
  left: 0;
  width: 100vw;
  height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  background: #333333;
  transition: opacity 2s, visibility 2s;
 }

 .loader--hidden {
  opacity: 0;
  visibility: hidden;
 }



 @keyframes loading {
  from {
   transform: rotate(0turn);
  }
  to {
   transform: rotate(1turn);
  }
 }

@keyframes dot1 {
 0% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #1e574f;
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #1e3f57;
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot2 {
  0% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #449180;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #447891;
    box-shadow: inset 0 0 15px 0 rgba(255, 255, 255, 0.2);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}

@keyframes dot3 {
  0% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }

  50% {
    border-color: #6bcdb2;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.8);
    transform: rotateX(20deg) rotateY(20deg) rotateZ(50deg) translateZ(0px);
  }

  100% {
    border-color: #6bb2cd;
    box-shadow: inset 0 0 15px 0 rgba(0, 0, 0, 0.1);
    transform: rotateX(24deg) rotateY(20deg) rotateZ(0deg) translateZ(-25px);
  }
}
.back-btn:hover {
  background-color: #fff !important;
  color: #ff5252 !important;
  border: 2px solid #ff5252 !important;
  box-shadow: 5px 5px 0px #ff5252 !important;
}
```

```
.back-btn:active {
 background-color: #fcf414 !important;
 box-shadow: none !important ;
 transform: translateY(4px) !important ;
}
.back-btn a{
 text-decoration: none !important ;
 color: white !important;
}
.back-btn a:hover{
 color: #ff5252 !important;
}
 @media screen and (max-width: 1000px) {
  body {
   align-items: flex-start;
   justify-content: flex-start;
  }
  .container1 {
   min-height: 100vh;
   flex-direction: column;
   border-radius: 0;
  }
  .container1 .left {
   width: 100%;
   height: 100%;
   padding: 20px 0;
  }
  .container1 .right {
   width: 100%;
   height: 100%;
   padding: 20px 0;
  }
  .calendar::before,
  .calendar::after {
   top: 100%;
   left: 50%;
   width: 97%;
   height: 12px;
   border-radius: 0 0 5px 5px;
   transform: translateX(-50%);
  }
  .calendar::before {
   width: 94%;
   top: calc(100% + 12px);
  }
  .events {
   padding-bottom: 340px;
  }
  .add-event-wrapper {
   bottom: 100px;
  }
 }
```

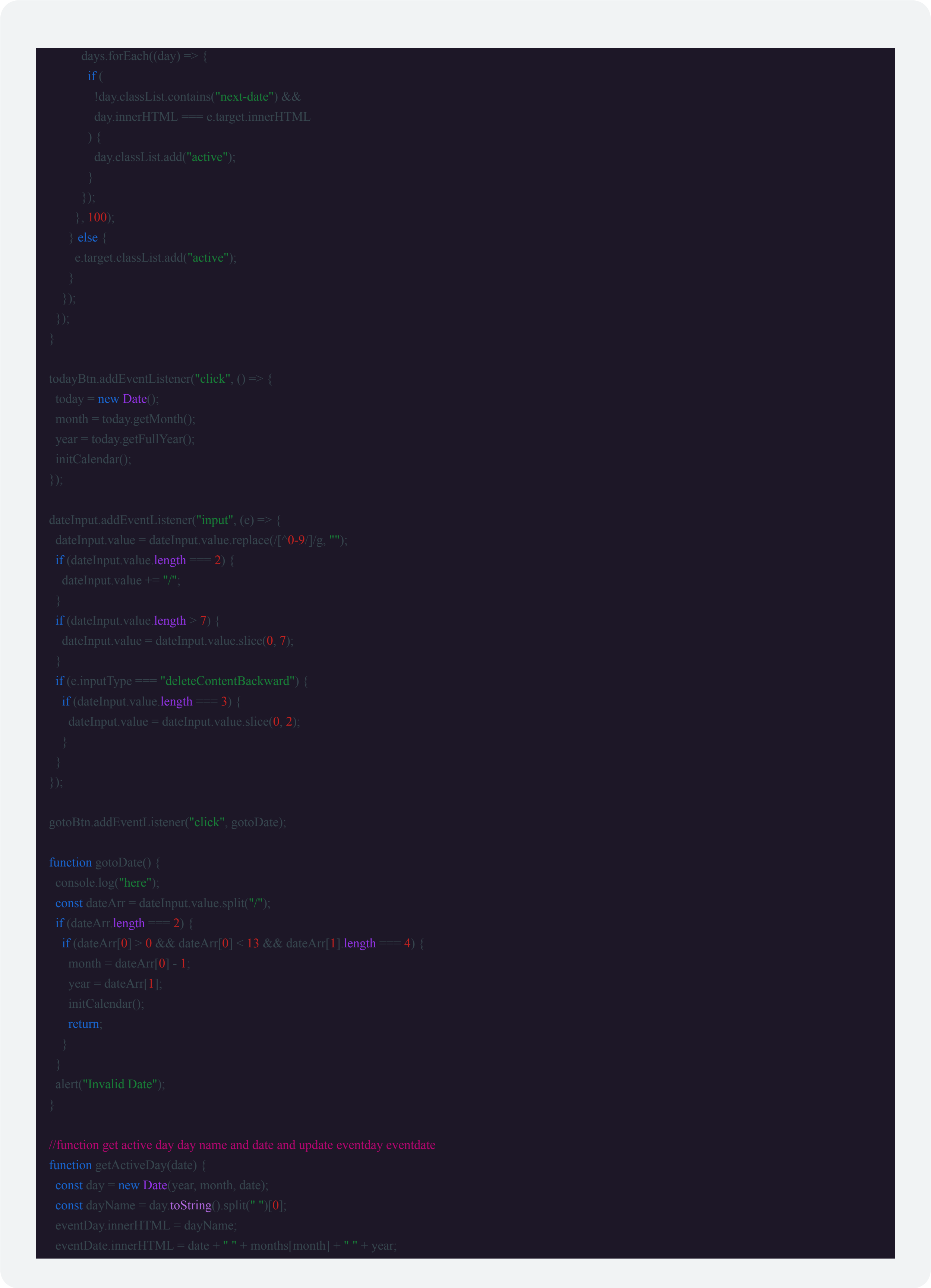
Calendar.js

JavaScript

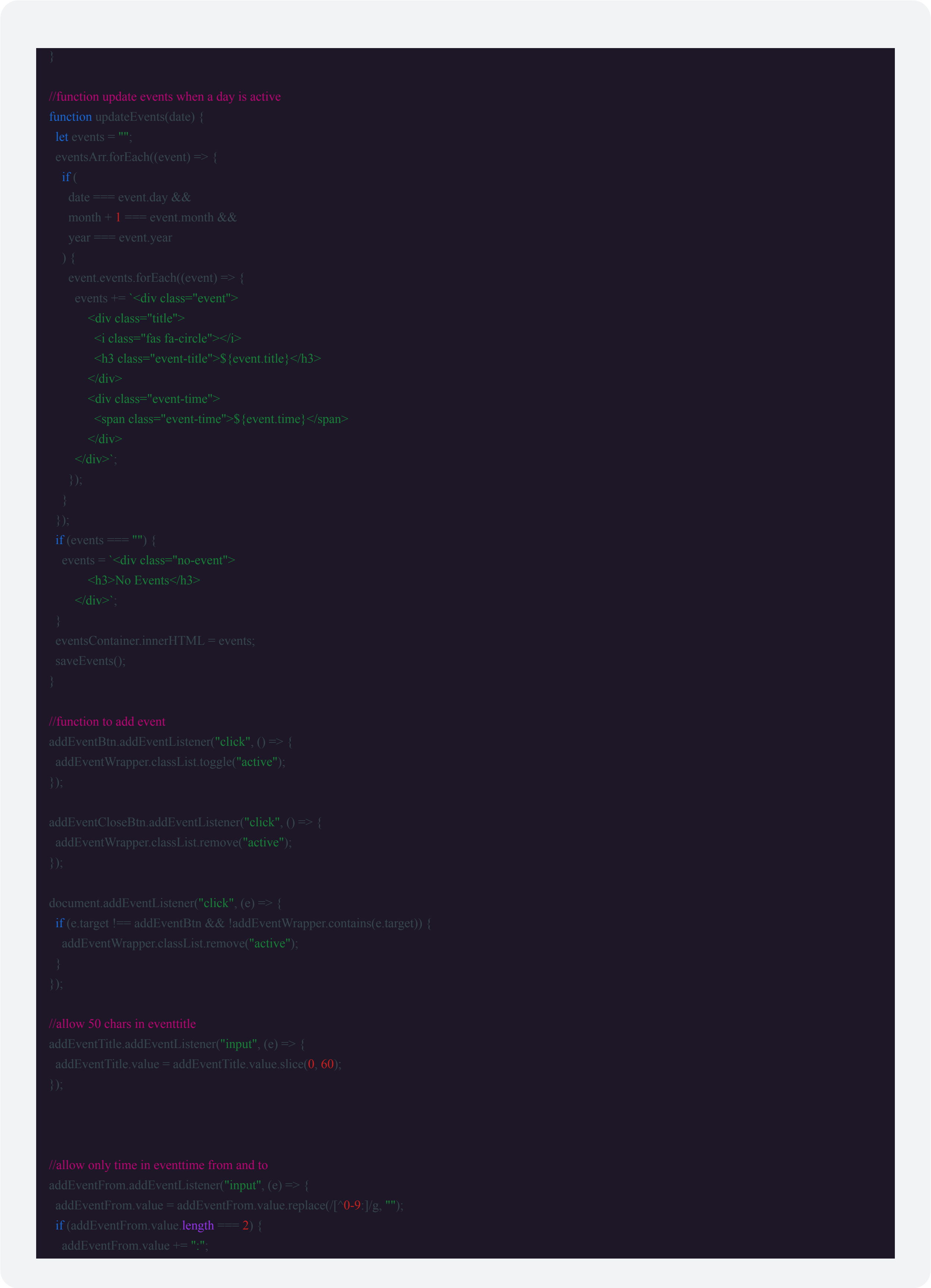
```javascript
document.addEventListener("DOMContentLoaded", function () {
  window.addEventListener("load", () => {
    const loader = document.querySelector(".loader");

    loader.classList.add("loader--hidden");

    loader.addEventListener("transitionend", () => {
     document.body.removeChild(loader);
    });
    });
  window.addEventListener("load", () => {
   const loader = document.querySelector(".loader");

   loader.classList.add("loader--hidden");

   loader.addEventListener("transitionend", () => {
    document.body.removeChild(loader);
   });
  });

  const calendar = document.querySelector(".calendar"),
   date = document.querySelector(".date"),
   daysContainer = document.querySelector(".days"),
   prev = document.querySelector(".prev"),
   next = document.querySelector(".next"),
   todayBtn = document.querySelector(".today-btn"),
   gotoBtn = document.querySelector(".goto-btn"),
   dateInput = document.querySelector(".date-input"),
   eventDay = document.querySelector(".event-day"),
   eventDate = document.querySelector(".event-date"),
   eventsContainer = document.querySelector(".events"),
   addEventBtn = document.querySelector(".add-event"),
   addEventWrapper = document.querySelector(".add-event-wrapper "),
   addEventCloseBtn = document.querySelector(".close "),
   addEventTitle = document.querySelector(".event-name "),
   addEventFrom = document.querySelector(".event-time-from "),
   addEventTo = document.querySelector(".event-time-to "),
   addEventSubmit = document.querySelector(".add-event-btn ");

  let today = new Date();
  let activeDay;
  let month = today.getMonth();
  let year = today.getFullYear();

  const months = [
   "January",
   "February",
   "March",
   "April",
   "May",
   "June",
   "July",
   "August",
   "September",
   "October",
   "November",
  "December",
];


const eventsArr = [];
getEvents();
console.log(eventsArr);

function initCalendar() {
  const firstDay = new Date(year, month, 1);
  const lastDay = new Date(year, month + 1, 0);
  const prevLastDay = new Date(year, month, 0);
  const prevDays = prevLastDay.getDate();
  const lastDate = lastDay.getDate();
  const day = firstDay.getDay();
  const nextDays = 7 - lastDay.getDay() - 1;

  date.innerHTML = months[month] + " " + year;

  let days = "";

  for (let x = day; x > 0; x--) {
    days += `<div class="day prev-date">${prevDays - x + 1}</div>`;
  }

  for (let i = 1; i <= lastDate; i++) {
    let event = false;
    eventsArr.forEach((eventObj) => {
      if (
        eventObj.day === i &&
        eventObj.month === month + 1 &&
        eventObj.year === year
      ) {
        event = true;
      }
    });
    if (
      i === new Date().getDate() &&
      year === new Date().getFullYear() &&
      month === new Date().getMonth()
    ) {
      activeDay = i;
      getActiveDay(i);
      updateEvents(i);
      if (event) {
        days += `<div class="day today active event">${i}</div>`;
      } else {
        days += `<div class="day today active">${i}</div>`;
      }
    } else {
      if (event) {
        days += `<div class="day event">${i}</div>`;
      } else {
        days += `<div class="day ">${i}</div>`;
      }
    }
  }

  for (let j = 1; j <= nextDays; j++) {
    days += `<div class="day next-date">${j}</div>`;
```



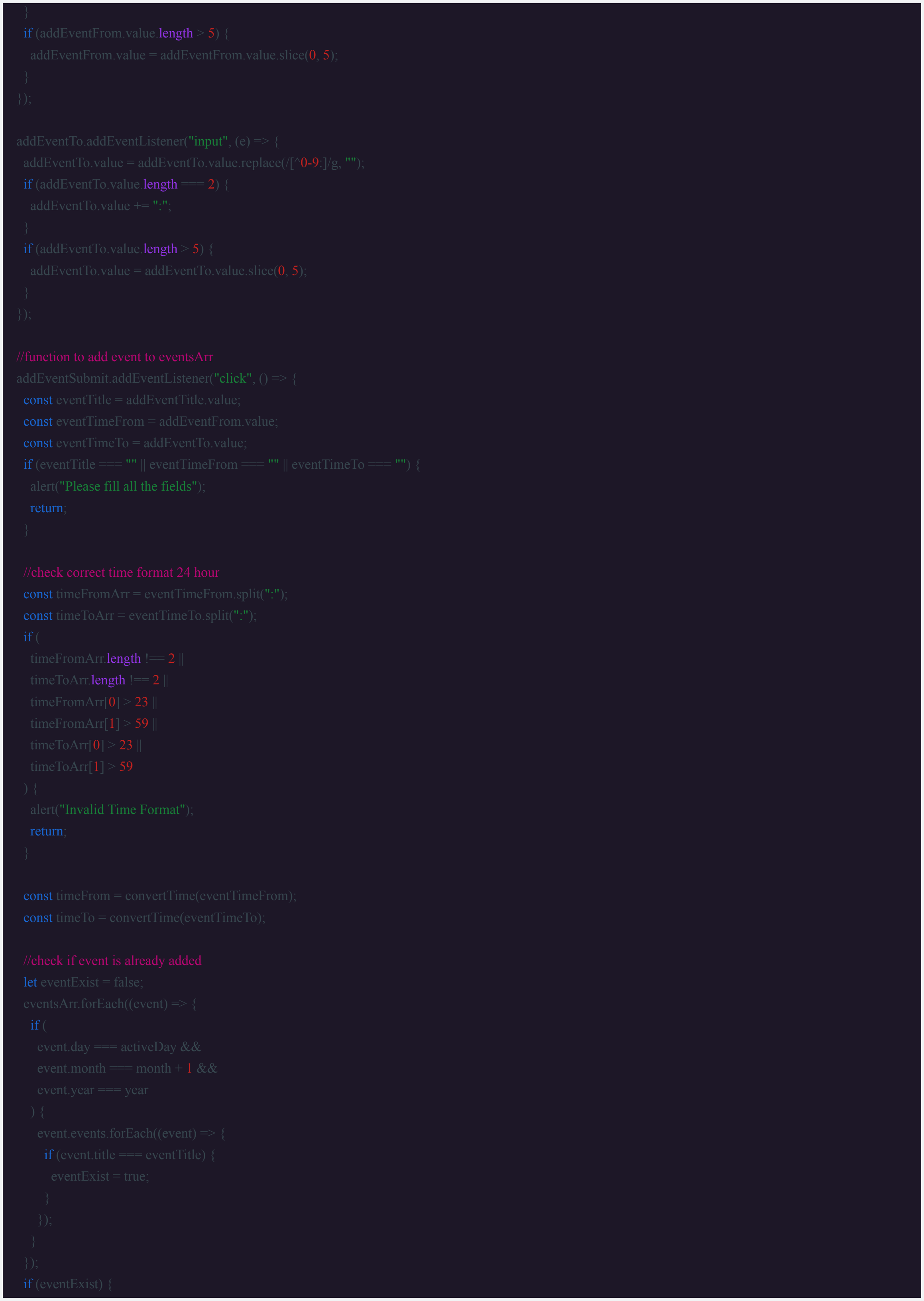
```
  }
  daysContainer.innerHTML = days;
  addListner();
}

function prevMonth() {
  month--;
  if (month < 0) {
    month = 11;
    year--;
  }
  initCalendar();
}

function nextMonth() {
  month++;
  if (month > 11) {
    month = 0;
    year++;
  }
  initCalendar();
}

prev.addEventListener("click", prevMonth);
next.addEventListener("click", nextMonth);

initCalendar();

function addListner() {
  const days = document.querySelectorAll(".day");
  days.forEach((day) => {
    day.addEventListener("click", (e) => {
      getActiveDay(e.target.innerHTML);
      updateEvents(Number(e.target.innerHTML));
      activeDay = Number(e.target.innerHTML);
      //remove active
      days.forEach((day) => {
        day.classList.remove("active");
      });
      //if clicked prev-date or next-date switch to that month
      if (e.target.classList.contains("prev-date")) {
        prevMonth();
        //add active to clicked day afte month is change
        setTimeout(() => {
          //add active where no prev-date or next-date
          const days = document.querySelectorAll(".day");
          days.forEach((day) => {
            if (
              !day.classList.contains("prev-date") &&
              day.innerHTML === e.target.innerHTML
            ) {
              day.classList.add("active");
            }
          });
        }, 100);
      } else if (e.target.classList.contains("next-date")) {
        nextMonth();
        //add active to clicked day afte month is changed
        setTimeout(() => {
          const days = document.querySelectorAll(".day");
```



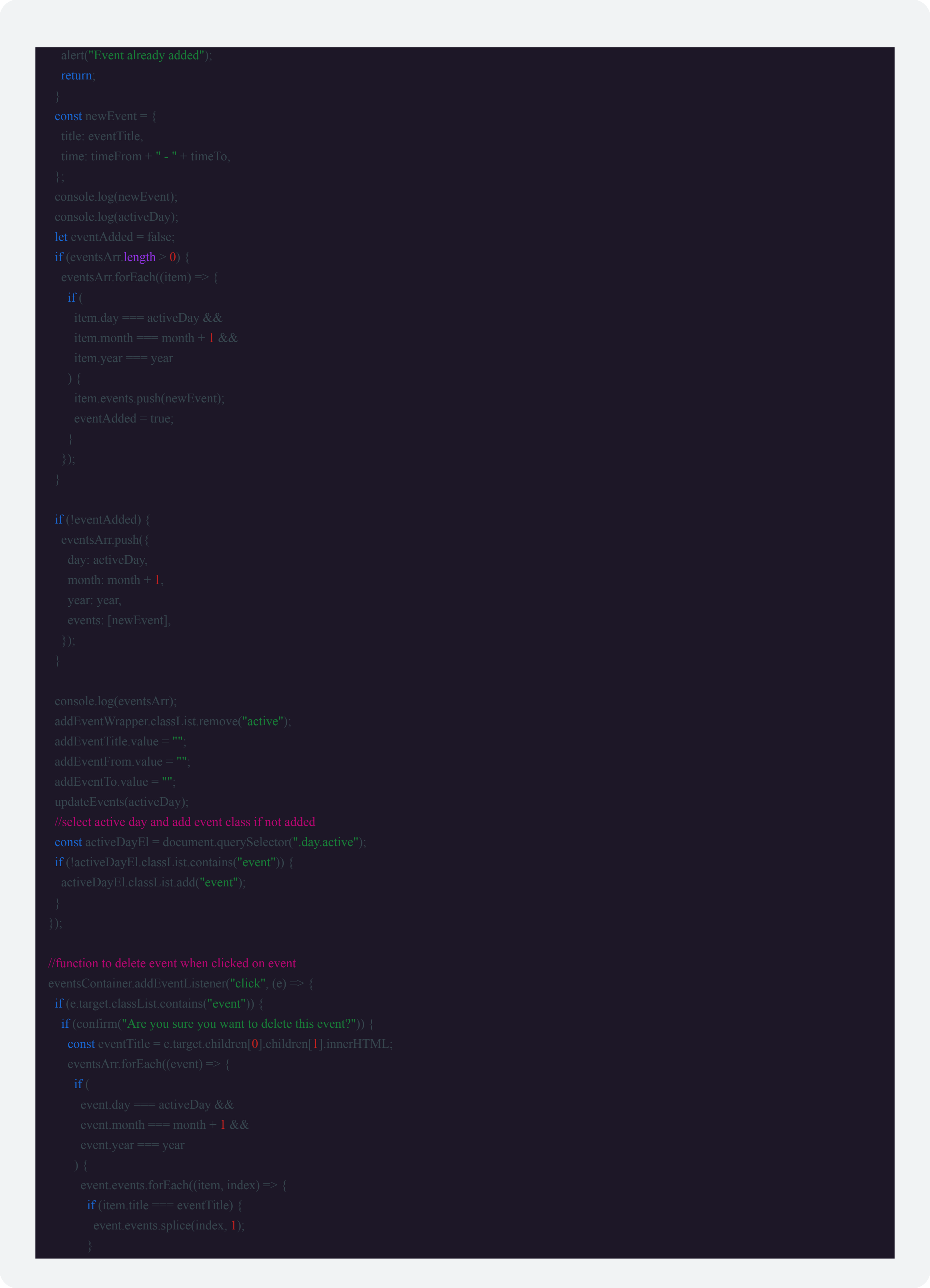
```
          days.forEach((day) => {
            if (
              !day.classList.contains("next-date") &&
              day.innerHTML === e.target.innerHTML
            ) {
              day.classList.add("active");
            }
          });
        }, 100);
      } else {
        e.target.classList.add("active");
      }
    });
  });
}

todayBtn.addEventListener("click", () => {
  today = new Date();
  month = today.getMonth();
  year = today.getFullYear();
  initCalendar();
});

dateInput.addEventListener("input", (e) => {
  dateInput.value = dateInput.value.replace(/[^0-9/]/g, "");
  if (dateInput.value.length === 2) {
    dateInput.value += "/";
  }
  if (dateInput.value.length > 7) {
    dateInput.value = dateInput.value.slice(0, 7);
  }
  if (e.inputType === "deleteContentBackward") {
    if (dateInput.value.length === 3) {
      dateInput.value = dateInput.value.slice(0, 2);
    }
  }
});

gotoBtn.addEventListener("click", gotoDate);

function gotoDate() {
  console.log("here");
  const dateArr = dateInput.value.split("/");
  if (dateArr.length === 2) {
    if (dateArr[0] > 0 && dateArr[0] < 13 && dateArr[1].length === 4) {
      month = dateArr[0] - 1;
      year = dateArr[1];
      initCalendar();
      return;
    }
  }
  alert("Invalid Date");
}

//function get active day day name and date and update eventday eventdate
function getActiveDay(date) {
  const day = new Date(year, month, date);
  const dayName = day.toString().split(" ")[0];
  eventDay.innerHTML = dayName;
  eventDate.innerHTML = date + " " + months[month] + " " + year;
```



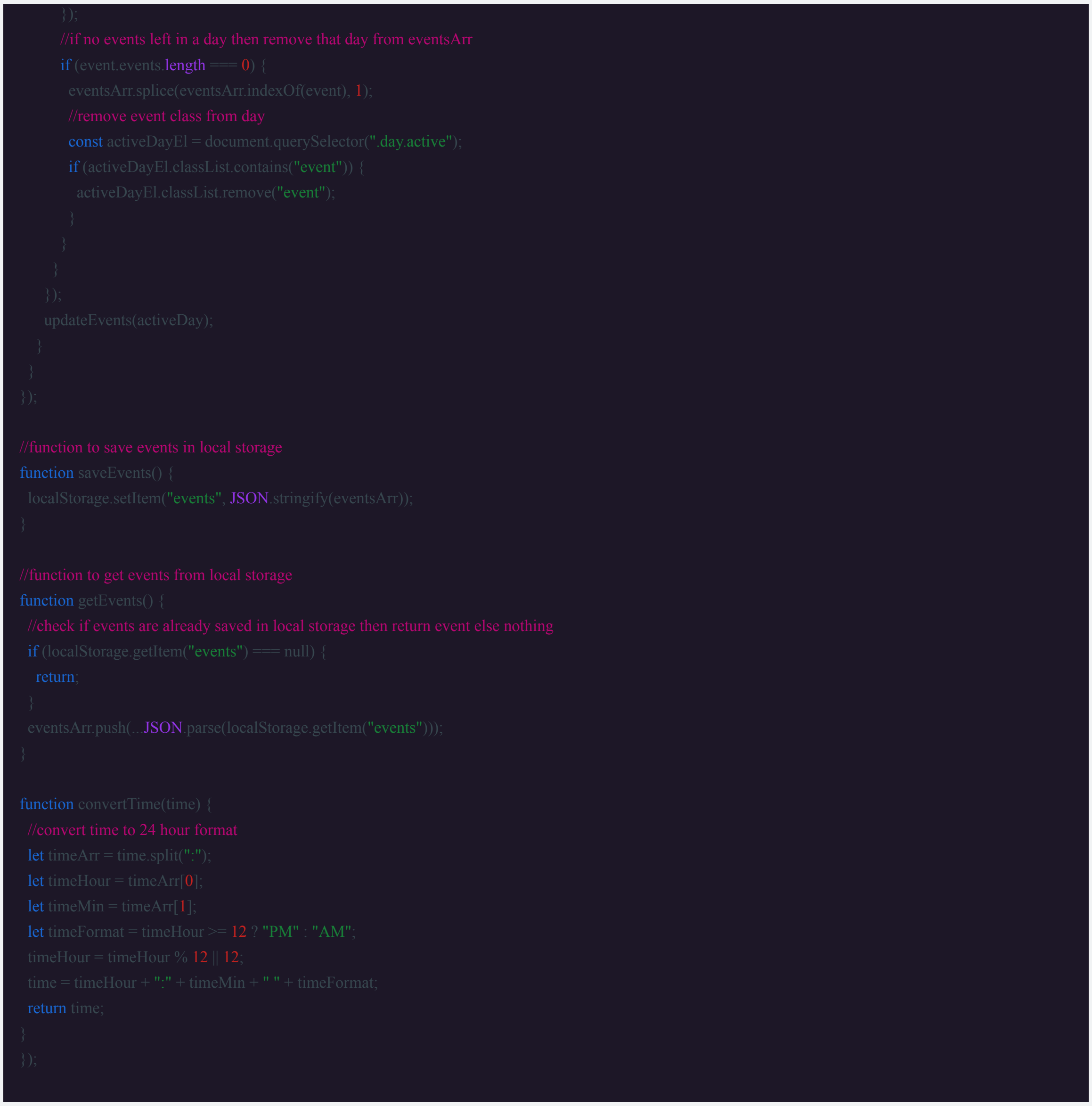
```
}

//function update events when a day is active
function updateEvents(date) {
  let events = "";
  eventsArr.forEach((event) => {
    if (
      date === event.day &&
      month + 1 === event.month &&
      year === event.year
    ) {
      event.events.forEach((event) => {
        events += `<div class="event">
            <div class="title">
              <i class="fas fa-circle"></i>
              <h3 class="event-title">${event.title}</h3>
            </div>
            <div class="event-time">
              <span class="event-time">${event.time}</span>
            </div>
        </div>`;
      });
    }
  });
  if (events === "") {
    events = `<div class="no-event">
            <h3>No Events</h3>
        </div>`;
  }
  eventsContainer.innerHTML = events;
  saveEvents();
}

//function to add event
addEventBtn.addEventListener("click", () => {
  addEventWrapper.classList.toggle("active");
});

addEventCloseBtn.addEventListener("click", () => {
  addEventWrapper.classList.remove("active");
});

document.addEventListener("click", (e) => {
  if (e.target !== addEventBtn && !addEventWrapper.contains(e.target)) {
    addEventWrapper.classList.remove("active");
  }
});

//allow 50 chars in eventtitle
addEventTitle.addEventListener("input", (e) => {
  addEventTitle.value = addEventTitle.value.slice(0, 60);
});



//allow only time in eventtime from and to
addEventFrom.addEventListener("input", (e) => {
  addEventFrom.value = addEventFrom.value.replace(/[^0-9:]/g, "");
  if (addEventFrom.value.length === 2) {
    addEventFrom.value += ":";
```



```
  }
  if (addEventFrom.value.length > 5) {
    addEventFrom.value = addEventFrom.value.slice(0, 5);
  }
});

addEventTo.addEventListener("input", (e) => {
  addEventTo.value = addEventTo.value.replace(/[^0-9:]/g, "");
  if (addEventTo.value.length === 2) {
    addEventTo.value += ":";
  }
  if (addEventTo.value.length > 5) {
    addEventTo.value = addEventTo.value.slice(0, 5);
  }
});

//function to add event to eventsArr
addEventSubmit.addEventListener("click", () => {
  const eventTitle = addEventTitle.value;
  const eventTimeFrom = addEventFrom.value;
  const eventTimeTo = addEventTo.value;
  if (eventTitle === "" || eventTimeFrom === "" || eventTimeTo === "") {
    alert("Please fill all the fields");
    return;
  }

  //check correct time format 24 hour
  const timeFromArr = eventTimeFrom.split(":");
  const timeToArr = eventTimeTo.split(":");
  if (
    timeFromArr.length !== 2 ||
    timeToArr.length !== 2 ||
    timeFromArr[0] > 23 ||
    timeFromArr[1] > 59 ||
    timeToArr[0] > 23 ||
    timeToArr[1] > 59
  ) {
    alert("Invalid Time Format");
    return;
  }

  const timeFrom = convertTime(eventTimeFrom);
  const timeTo = convertTime(eventTimeTo);

  //check if event is already added
  let eventExist = false;
  eventsArr.forEach((event) => {
    if (
      event.day === activeDay &&
      event.month === month + 1 &&
      event.year === year
    ) {
      event.events.forEach((event) => {
        if (event.title === eventTitle) {
          eventExist = true;
        }
      });
    }
  });
  if (eventExist) {
```



```
    alert("Event already added");
    return;
  }
  const newEvent = {
    title: eventTitle,
    time: timeFrom + " - " + timeTo,
  };
  console.log(newEvent);
  console.log(activeDay);
  let eventAdded = false;
  if (eventsArr.length > 0) {
    eventsArr.forEach((item) => {
      if (
        item.day === activeDay &&
        item.month === month + 1 &&
        item.year === year
      ) {
        item.events.push(newEvent);
        eventAdded = true;
      }
    });
  }

  if (!eventAdded) {
    eventsArr.push({
      day: activeDay,
      month: month + 1,
      year: year,
      events: [newEvent],
    });
  }

  console.log(eventsArr);
  addEventWrapper.classList.remove("active");
  addEventTitle.value = "";
  addEventFrom.value = "";
  addEventTo.value = "";
  updateEvents(activeDay);
  //select active day and add event class if not added
  const activeDayEl = document.querySelector(".day.active");
  if (!activeDayEl.classList.contains("event")) {
    activeDayEl.classList.add("event");
  }
});

//function to delete event when clicked on event
eventsContainer.addEventListener("click", (e) => {
  if (e.target.classList.contains("event")) {
    if (confirm("Are you sure you want to delete this event?")) {
      const eventTitle = e.target.children[0].children[1].innerHTML;
      eventsArr.forEach((event) => {
        if (
          event.day === activeDay &&
          event.month === month + 1 &&
          event.year === year
        ) {
          event.events.forEach((item, index) => {
            if (item.title === eventTitle) {
              event.events.splice(index, 1);
            }
```



```
          });
          //if no events left in a day then remove that day from eventsArr
          if (event.events.length === 0) {
            eventsArr.splice(eventsArr.indexOf(event), 1);
            //remove event class from day
            const activeDayEl = document.querySelector(".day.active");
            if (activeDayEl.classList.contains("event")) {
              activeDayEl.classList.remove("event");
            }
          }
        }
      });
      updateEvents(activeDay);
    }
  }
});

//function to save events in local storage
function saveEvents() {
  localStorage.setItem("events", JSON.stringify(eventsArr));
}

//function to get events from local storage
function getEvents() {
  //check if events are already saved in local storage then return event else nothing
  if (localStorage.getItem("events") === null) {
    return;
  }
  eventsArr.push(...JSON.parse(localStorage.getItem("events")));
}

function convertTime(time) {
  //convert time to 24 hour format
  let timeArr = time.split(":");
  let timeHour = timeArr[0];
  let timeMin = timeArr[1];
  let timeFormat = timeHour >= 12 ? "PM" : "AM";
  timeHour = timeHour % 12 || 12;
  time = timeHour + ":" + timeMin + " " + timeFormat;
  return time;
}
});
```