



папка с публично доступными файлами

папка для хранения картинок отправленных
пользователями

папка с доп функциями для работы
сайта

стили

шаблоны страниц

код сервера

index.hbs (today.hbs фактически аналогична)

```
22 <body>
23   <div class="left-menu">
24     <h2><a href="/profile"><svg
25       xmlns="http://www.w3.org/2000/svg"
26       width="22"
27       height="22"
28       fill="currentColor"
29       class="bi bi-gear-fill"
30       viewBox="0 0 16 16"
31     >
32       <path
33         d="M9.405 1.05c-.413-1.4-2.397-1.4-2.81 0l-.134a1.
34       />
35     </svg></a></h2>
36     <br />
37     <p><a href="/today">Today</a></p>
38     <p><a href="#">All posts</a></p>
39   </div>
40   <div class="content">
41     <h1>My Posts:</h1>
42     <div class="entries-container">
43       {{#if entries.length}}
44       {{#each entries}}
45         <div class="entry">
46           <p class="entry-time1">{{this.currentDate}}</p>
47           <div class="entry-text">
48             {{{this.text}}}
49           </div>
50           {{#if this.image}}
51             
55           {{/if}}
56         </div>
```

```
56   <div class="entry-menu">
57     <p class="entry-time">{{currentTime}}</p>
58     <div class="entry-menu-inner">
59       <button class="edit-button">Edit</button>
60       <button class="delete-button">Delete</button>
61     </div>
62   </div>
63 {{/each}}
64 {{else}}
65   <p>No posts.</p>
66 {{/if}}
67
68 <div class="formdiv">
69   <button id="emoji-button" onclick="toggleEmojiMenu()">Select Emoji</button>
70   <div id="emoji-menu" class="emoji-menu" style="display: none;">
71     <button class="emoji" onclick="addEmoji('😄')">😄</button>
72     <button class="emoji" onclick="addEmoji('😊')">😊</button>
73     <button class="emoji" onclick="addEmoji('👍')">👍</button>
74     <button class="emoji" onclick="addEmoji('❤️')">❤️</button>
75   </div>
76   <form
77     id="entry-form" action="/submit" method="post"
78     enctype="multipart/form-data">
79     <textarea name="text" id="entry-textarea" rows="7"
80       cols="50" required placeholder="What's new?"
81     ></textarea>
82     <br />
83     <input id="fileinput" type="file" name="image"
84       accept="image/*" placeholder="Choose file"
85     />
86     <br /><br />
87     <input type="submit" value="Post" id="inputsub" />
88   </form>
```

profile.hbs

```
23 <body>
24   <div class="left-menu">
25     <h2><a href="#"><svg
26       xmlns="http://www.w3.org/2000/svg"
27       width="22"
28       height="22"
29       fill="currentColor"
30       class="bi bi-gear-fill"
31       viewBox="0 0 16 16"
32     >
33       <path
34         d="M9.405 1.05c-.413-1.4-2.397-1.4-2.81 0l-.134a1.464 1.464 0
35       />
36     </svg></a></h2>
37   <br />
38   <p><a href="/today">Today</a></p>
39   <p><a href="/index">All posts</a></p>
40 </div>
41 <div class="settings-main">
42   <h1>Profile settings</h1>
43   <hr />
44   <div class="settings-full-info">
45     <div class="settings-info">
46       <div class="settings-name">
47         <h2>Your name</h2>
48         <input type="text" value="{{user.username}}" readonly />
49       </div>
50     </div>
51     <div class="settings-email">
52       <hr />
53       <h2>Email</h2>
54       <div class="inner-settings-email">
55         <div class="inner-settings-email-first">
56           <input
```

```
57     type="text"
58     class="input-inner-settings-email"
59     value="{{user.email}}"
60     readonly
61   />
62 </div>
63 <div class="inner-settings-email-last">
64   <input
65     type="checkbox"
66     class="form-check-input"
67     id="settings-verify-email"
68     checked
69     readonly
70   />
71   <label for="settings-verify-email">Verified</label>
72 </div>
73 </div>
74 <hr />
75 </div>
76 <div class="changeuser">
77   <a href="/logout">Log out</a>
78 </div>
79 </div>
80 </div>
81 </body>
82
83 </html>
```

register.hbs

```
8 <link rel="stylesheet" type="text/css" href="/styles/styles_register."
9 </head>
10
11 <body>
12   <div class="container">
13     <div class="form-container">
14       <h1>Registration</h1>
15       {{#if error}}
16       <p style="color: red;">{{error}}</p>
17       {{/if}}
18       <form action="/register" method="post">
19         <div class="form-group">
20           <label for="username">Username:</label>
21           <input type="text" id="username" name="username" required />
22         </div>
23         <div class="form-group">
24           <label for="email">Email:</label>
25           <input type="email" id="email" name="email" required />
26         </div>
27         <div class="form-group">
28           <label for="password">Password:</label>
29           <input type="password" id="password" name="password"
30             required />
31         </div>
32         <button type="submit">Register</button>
33       </form>
34       <p>Already have an account? <a href="/login">Login</a></p>
35     </div>
36   </div>
37 </body>
38
39 </html>
```

login.hbs

```
9 </head>
10
11 <body>
12   <div class="container">
13     <div class="form-container">
14       <h1>Login</h1>
15       {{#if error}}
16       <p style="color: red;">{{error}}</p>
17       {{/if}}
18       <form action="/login" method="post">
19         <div class="form-group">
20           <label for="username">Username:</label>
21           <input type="text" id="username" name="username" required />
22         </div>
23         <div class="form-group">
24           <label for="email">Email:</label>
25           <input type="email" id="email" name="email" required />
26         </div>
27         <div class="form-group">
28           <label for="password">Password:</label>
29           <input type="password" id="password" name="password" required />
30         </div>
31         <button type="submit">Login</button>
32       </form>
33       <p>Don't have an account? <a href="/">Register</a></p>
34     </div>
35   </div>
36 </body>
37
38 </html>
```

logout.hbs

```
<body>
  <div class="overlay" id="overlay"></div>
  <div class="modal" id="modal">
    <p>Are you sure you want to log out?</p>
    <button id="confirm">Yes</button>
    <button id="cancel">Cancel</button>
  </div>
</body>
<script>
  // Отображение модального окна
  document.getElementById("overlay").style.display = "block";
  document.getElementById("modal").style.display = "block";
  // Обработчик события для кнопки подтверждения
  document.getElementById("confirm").addEventListener("click", function () {
    // Отправка запроса на сервер для завершения сеанса
    window.location.href = "/"; });
  // Обработчик события для кнопки отмены
  document.getElementById("cancel").addEventListener("click", function () {
    // Закрытие модального окна
    document.getElementById("overlay").style.display = "none";
    document.getElementById("modal").style.display = "none";
    // Возврат пользователя на предыдущую страницу
    window.history.back(); });
</script>
</html>
```

Registration

Username:

Email:

Password:

Register

Already have an account? [Login](#)

```
{/}
```

```
<form action="/register" method="post">
```

```
<div class="form-group">
```

```
app.post("/register", (req, res) => {  
  // Получение данных из формы регистрации  
  const { username, password, email } = req.body;  
  
  // Проверка наличия пользователя с таким же именем  
  if (users.has(username)) {  
    return res.render("register", {  
      error: "A user with this username already exists.",  
    });  
  }  
  
  // Генерация уникального идентификатора пользователя  
  const userId = generateUniqueId();  
  
  // Добавление нового пользователя в Map  
  users.set(username, { username, password, email, userId });  
  
  // Установка сессии для аутентификации  
  req.session.user = username;  
  req.session.userId = userId;  
  
  // Редирект на главную страницу  
  res.redirect("/index");  
});
```

Login

Username:

Email:

Password:

Login

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

```
<form action="/login" method="post">
```

```
app.post("/login", (req, res) => {  
  // Получение данных из формы аутентификации  
  const { username, password, email } = req.body;  
  
  // Проверка соответствия логина и пароля  
  if (users.has(username) && users.get(username).password === password  
    && users.get(username).email === email) {  
    req.session.user = username;  
    req.session.userId = users.get(username).userId;  
    return res.redirect("/index");  
  } else {  
    return res.render("login", { error: "Incorrect username, email or password." });  
  }  
});
```

My Posts:

16 January 2024

sxaxasxas

15:29

Edit

Delete

Select Emoji

fwfwfwefew

Выберите файл

Файл не выбран

2.

1.

Post

1.

```
app.post("/submit", checkAuth, (req, res) => {
  // Обработка формы создания записи
  const currentDate = moment().format("DD MMMM YYYY");
  const text = req.body.text;
  const currentTime = moment().format("HH:mm");

  let imageUrl = null;
  // Обработка загруженного изображения
  if (req.files && req.files.image) {
    const image = req.files.image;
    const uniqueSuffix = `${Date.now()}-${Math.round(Math.random() * 1e9)}`;
    const imageName = `${uniqueSuffix}-${image.name}`;
    imageUrl = `/images/${imageName}`;

    // Сохранение изображения в папке public/images
    image.mv(`public/images/${imageName}`, (err) => {
      if (err) {
        console.error(err);
      }
    });
  }

  // Добавление новой записи в массив
  entries.push({
    currentDate,
    text,
    currentTime,
    image: imageUrl,
    author: req.session.user,
  });

  // Редирект на главную страницу
  res.redirect("/index");
});
```

2.

```
<input id="fileinput" type="file" name="image"
| accept="image/*" placeholder="Choose file"
/>
```

```
let imageUrl = null;
// Обработка загруженного изображения
if (req.files && req.files.image) {
  const image = req.files.image;
  const uniqueSuffix = `${Date.now()}-${Math.round(Math.random() * 1e9)}`;
  const imageName = `${uniqueSuffix}-${image.name}`;
  imageUrl = `/images/${imageName}`;

  // Сохранение изображения в папке public/images
  image.mv(`public/images/${imageName}`, (err) => {
    if (err) {
      console.error(err);
    }
  });
}
```

```
<form
id="entry-form" action="/submit" method="post"
enctype="multipart/form-data">
```

My Posts:

16 January 2024

sxaxasxas

15:29

Edit

Delete

1.

3.

2.

Select Emoji

fwfwfwefew

Выберите файл Файл не выбран

Post

1, 2

```
<button class="edit-button">Edit</button>
<button class="delete-button">Delete</button>
```

```
document.addEventListener("DOMContentLoaded", function () {
  // Получаем коллекции кнопок редактирования и удаления
  const editButtons = document.querySelectorAll(".edit-button");
  const deleteButtons = document.querySelectorAll(".delete-button");

  // Добавляем обработчик события для каждой кнопки редактирования
  editButtons.forEach(function (button, index) {
    button.addEventListener("click", function () {
      // Вызываем функцию редактирования с передачей индекса записи
      editEntry(index);
    });
  });

  // Добавляем обработчик события для каждой кнопки удаления
  deleteButtons.forEach(function (button, index) {
    button.addEventListener("click", function () {
      // Вызываем функцию удаления с передачей индекса записи
      deleteEntry(index);
    });
  });
});
```

1.

```
// Функция редактирования записи
function editEntry(index) {
    // Получаем текстовое поле записи по индексу
    const entryText = document.querySelectorAll(".entry-text")[index];
    // Запрашиваем у пользователя новый текст через диалоговое окно
    const newText = prompt("Введите новый текст", entryText.textContent);

    // Проверяем, был ли введен новый текст
    if (newText !== null) {
        // Обновляем текст записи на странице
        entryText.textContent = newText;

        // Отправляем запрос на сервер для обновления текста записи
        fetch(`/edit/${index}`, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({ text: newText }),
        })
        .then(function (response) {
            // Проверяем успешность запроса и выводим соответствующее сообщение
            if (response.ok) {
                alert("Запись успешно отредактирована!");
            } else {
                // В случае ошибки выводим сообщение об ошибке
                throw new Error("Ошибка редактирования записи :(");
            }
        })
        .catch(function (error) {
            // Выводим сообщение об ошибке в консоль и в окно оповещения
            console.error(error);
            alert("Ошибка");
        });
    }
}
```

script.js

2.

```
// Функция удаления записи
function deleteEntry(index) {
    // Проверяем у пользователя подтверждение на удаление записи
    if (confirm("Вы уверены, что хотите удалить запись?")) {
        // Отправляем запрос на сервер для удаления записи
        fetch(`/delete/${index}`, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
        })
        .then(function (response) {
            // Проверяем успешность запроса и перезагружаем страницу в случае успеха
            if (response.ok) {
                window.location.reload();
            } else {
                // В случае ошибки выводим сообщение об ошибке
                throw new Error("Ошибка удаления записи :(");
            }
        })
        .catch(function (error) {
            // Выводим сообщение об ошибке в консоль и в окно оповещения
            console.error(error);
            alert("Ошибка");
        });
    }
}
```

3.

```
<button id="emoji-button" onclick="toggleEmojiMenu()">Select Emoji</button>
```

entries.js

```
// Функция для отображения/скрытия меню с эмодзи
function toggleEmojiMenu() {
    // Получение элемента с меню эмодзи
    const emojiMenu = document.getElementById("emoji-menu");

    // Установка стиля display в "block" для отображения меню
    emojiMenu.style.display = "block";
}

// Функция для добавления выбранного эмодзи в текстовое поле
function addEmoji(emoji) {
    // Получение элемента текстового поля
    const textarea = document.getElementById("entry-textarea");

    // Добавление выбранного эмодзи к текущему содержимому текстового поля
    textarea.value += emoji;
}
```

Are you sure you want to log out?

Yes

Cancel

```
<script>
```

```
// Отображение модального окна
```

```
document.getElementById("overlay").style.display = "block";
```

```
document.getElementById("modal").style.display = "block";
```

```
// Обработчик события для кнопки подтверждения
```

```
document.getElementById("confirm").addEventListener("click", function () {
```

```
// Отправка запроса на сервер для завершения сеанса
```

```
window.location.href = "/"; });
```

```
// Обработчик события для кнопки отмены
```

```
document.getElementById("cancel").addEventListener("click", function () {
```

```
// Закрытие модального окна
```

```
document.getElementById("overlay").style.display = "none";
```

```
document.getElementById("modal").style.display = "none";
```

```
// Возврат пользователя на предыдущую страницу
```

```
window.history.back(); });
```