1. Zapis ról przydzielonych w grupie projektowej.

Tworzenie warstwy front-end

1. Opis tworzonych funkcjonalności

Stworzenie strony logowania, z której później zrezygnowano.

Zaprojektowanie tła aplikacji.

Stworzenie strony profilu użytkownika.

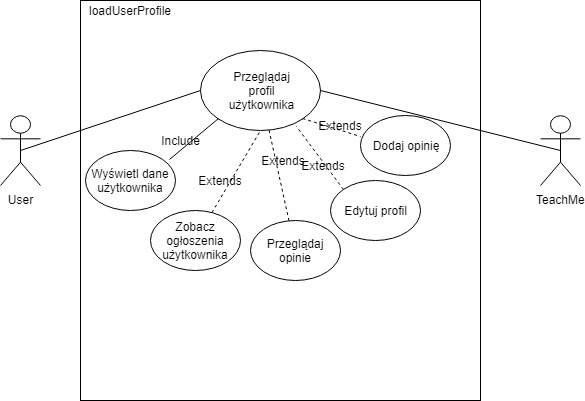
Stworzenie strony edycji profilu użytkownika.

Stworzenie kart opinii użytkownika.

Dokonanie zmian w stronie ogłoszenia.

Dokonanie zmian w działaniu menu głównego.

1. Diagram przypadku Użycia funkcjonalności

Profil użytkownika

1. Omówienie stworzonego kodu i jego działania
2. function loadUserProfile(idUser) {
3. let user;
4. //utworzenie requestu
5. let request = new XMLHttpRequest();
6. if (idUser != 0 && idUser != "undefined") {
7. //inicjalizacja rządania
8. request.open('GET', usersUrl + '/' + idUser, false);
9. //request pomyśłny
10. request.onload = function () {
11. // Dostęp do danych z JSON
12. user = JSON.parse(this.response);
13. // Request poprawny
14. if (request.status >= 200 && request.status < 400) {
15. // stworzenie obiektu user z danymi z JSON
16. user = new User(user.idUser, user.login, user.name, user.surname,
17. user.birthDate, user.avatar, user.phone, user.email,
18. user.cityByCityIdCity.name, user.cityByCityIdCity.idCity, user.about);
19. } else {
20. console.log('error');
21. }
22. // załadowanie zawartości userProfile do zmiennej
23. const userHTML = document.getElementById('userProfile');
24. //skrócony profil
25. html = '';
26. html += '<div class="card-body">';
27. html += '<h4 class="card-title profile-name">' + user.name + ' ' + user.surname + '</h4></div>';
28. html += '<img class="avatar" src="' + user.avatar + '" alt="Card image">';
29. html += '<div class="card-body">';
30. html += '<h5 class="card-title">O mnie:</h5>'
31. html += '<p class="card-text">' + user.about + '</p>';
32. html += '<span class="badge badge-info opinionsBtn" id="opinionAmount" onclick="scrollDown(); storeUserId(' + user.idUser + ');"></span>';
33. html += '<span class="badge badge-warning opinionsBtn" id="ratingAvg"></span>';
34. // pełny profil
35. if (window.location.pathname.substr(-12) === 'profile.html' || window.location.pathname.substr(-14) === 'profilemy.html') {
36. html += '<ul class="list-group list-group-flush" style="font-size: 20px;">';
37. html += '<li class="list-group-item"><i class="material-icons">phone</i> ' + user.phone + '</li>';
38. html += '<li class="list-group-item"><i class="material-icons">email</i> ' + user.email + '</li>';
39. html += '<li class="list-group-item"><i class="material-icons">location\_city</i> ' + user.cityName + '</li>';
40. html += '</ul>';
41. html += '</div>';
42. html += '<button type="button" class="btn btn-success show-ann" id="showNoticesBtn" style="margin-bottom:5px" onclick="loadUserNotices(' + user.idUser + ')">Zobacz ogłoszenia</button>';
43. html += '<button type="button" class="btn btn-success show-ann" id="showOpinionBtn" style="margin-bottom:5px" onclick="loadUserOpinions(' + user.idUser + ')">Zobacz opinie</button>';
44. // jeżeli użytkownik przegląda swój profil
45. if (Number(localStorage.getItem("loggedID")) === user.idUser) {
46. html += '<a href="profileedit.html"><button class="circleButton editButton"><i class="material-icons">edit</i></button></a>'
47. } else {
48. html += '<a href="addopinion.html"><button type="button" class="btn btn-success show-ann">Dodaj opinię</button></a>';
49. }
50. } else {
51. html += '</div>';
52. html += '<a href="profile.html"><button type="button" class="btn btn-success show-ann" id="btnShowProfile"">Zobacz Profil</button></a>';
53. }
54. // podmiana zawartości
55. userHTML.innerHTML = html;
56. };
57. //zakończ request
58. request.send();
59. }
60. }
61. Podsumowanie

Praca indywidualna polegała na pisaniu kodu z zakresu front-endu, więc korzystano z JavaScript, CSS, HTML. Większość zadań opierała się na zaprojektowaniu wyglądu i zaimplementowaniu go. Kolejnym krokiem było obsługiwanie plików JSON otrzymanych od serwera lub wysyłanie ich do serwera.