

Project Report: Enhancing Functionality and Overcoming Challenges

In this project, I developed a comprehensive system comprising various files and functionalities to manage staff members, their activities, and time-related operations. The project involves the creation of several files, each serving specific purposes to ensure a seamless operation of the system.

File Structure and Class Hierarchy

I initiated the project by creating a file named `class.js`, which houses distinct classes: “Employee”, “Staff_Member” as in figure 1, and “Delivery_Driver” as shown in figure 2. The latter two classes inherit properties and behaviors from the `Employee` class, facilitating a streamlined structure for managing different staff roles.

What this class containing are as shown in screenshot, picture, email, time the staff being out and status of staff. Additionally, the class contain a function called “StaffMemberIsLate” which triggers a toast notification if a member is late, providing comprehensive information about the tardiness.

```
10  }
11
12  class Staff_Member extends Employee {
13      picture;
14      email;
15      status;
16      outTime;
17      duration;
18      expectedReturnTime;
19
20      constructor(name, surname, picture, email, status, outTime, duration, expectedReturnTime) {
21          super(name, surname);
22          this.picture = picture;
23          this.email = email;
24          this.status = status;
25          this.outTime = outTime;
26          this.duration = duration;
27          this.expectedReturnTime = expectedReturnTime;
28      }
29
30      staffMemberIsLate() {
31          return ('Name: ' + this.name + '<br>' + 'Surname: ' + this.surname + '<br>' + 'Time away: ' + this.duration);
32      }
33  }
```

Figure 1 Staff_Member class

The Delivery_driver class contains following properties as vehicle, telephone, deliver Adress and return Time. Additionally, the class contain a function called “deliveryDriverIsLate” which triggers a toast notification if the delivery driver is late.

```

33 }
34
35 class Delivery_driver extends Employee {
36     vehicle;
37     telephone;
38     deliverAddress;
39     returnTime;
40
41     constructor(name, surname, vehicle, telephone, deliverAddress, returnTime) {
42         super(name, surname);
43         this.id = Date.now(); //generate a unique ID for the driver
44         this.vehicle = vehicle;
45         this.telephone = telephone;
46         this.deliverAddress = deliverAddress;
47         this.returnTime = returnTime;
48     }
49
50     deliveryDriverIsLate() {
51         return ('Full Name: ' + this.name + ' ' + this.surname + '<br>' + 'phone number: ' + this.telephone +
52             '<br>' + 'Was suppose to return: ' + this.returnTime +
53             '<br>' + 'Delivery address: ' + this.deliverAddress + '<br>' + '<br>');
54     }
55 }

```

Figure 2 Delivery_driver class

Time Management Module

The “Clock.js” file was designed to acquire and manage time-related data such as date, hour, and other relevant information. This module is crucial for various time-sensitive operations within the system.

It contain a function called “digitalClock”, to retrieve and display the current time (hours, minutes, seconds). It formats the time values to always show two digits and updates the HTML elements with the corresponding classes to reflect the current time. The “setInterval” method calls “digitClock” every second to keep the time updated in real-time. Additionally, it retrieves the current data and displays it in a specific format, updating the respective HTML elements with day, date, month and year information. See figure 3.

```

4  function digitalClock() {
5      var today = new Date();
6
7      var hours = today.getHours();
8      var minutes = today.getMinutes();
9      var seconds = today.getSeconds();
10
11     hours = hours > 24 ? hours % 24 : hours;
12
13
14     //Add the 0 for the values lower than 10
15     if (hours < 10) {
16         hours = "0" + hours;
17     }
18     if (minutes < 10) {
19         minutes = "0" + minutes;
20     }
21     if (seconds < 10) {
22         seconds = "0" + seconds;
23     }
24
25
26     document.querySelector(".hours").innerHTML = hours;
27     document.querySelector(".minutes").innerHTML = minutes;
28     document.querySelector(".seconds").innerHTML = seconds;
29
30 }
31
32 var updateClock = setInterval(digitalClock, 1000);
33
34 //Get the date
35 var today = new Date();
36 const dayNumber = today.getDate();
37 const year = today.getFullYear();
38 const dayName = today.toLocaleString("default", { weekday: "short" });
39 const monthName = today.toLocaleString("default", { month: "short" });
40
41
42 document.querySelector(".day-name").innerHTML = dayName;
43 document.querySelector(".day-number").innerHTML = dayNumber;
44 document.querySelector(".month-name").innerHTML = monthName;
45 document.querySelector(".year").innerHTML = year;
46

```

Figure 3 clock's contains

Delivery Management

Another integral component is the "Delivery.js" file, housing functions essential for managing deliveries. Functions like:

- 1- "deliveryDriverIsLate(driverId): this function checks if a delivery driver is late based on their return time. It continuously checks and triggers a toast notification if the driver is overdue.

- 2- “addDelivery”: validates input data for a new delivery, creates a table row displaying the details, highlights the added row, creates a new Delivery Driver object, triggers the “deliveryDriverIsLate” function for the added driver, and clears input fields after successful addition.
- 3- “validateDelivery”: validate the input criteria for a delivery such as phone number length. Presence of name, surname, address, and return date.
- 4- “deliveryClear”: asks for confirmation before removing a delivery. If confirmed, it empties the content of the highlighted delivery row. Otherwise, it removes the highlighting from the row.

See figure 4 down.

```

3      function deliveryDriverIsLate(driverId) {
4          let driver = drivers.find(d => d.id === driverId)
5          console.log(driver.returnTime)
6          if (driver.returnTime > new Date(new Date().getTime()).toLocaleTimeString()) {
7              setTimeout(function() {
8                  deliveryDriverIsLate(driver.id)
9              }, 1000);
10         } else if (driver.returnTime < new Date(new Date().getTime()).toLocaleTimeString()) {
11             $('#deliverToastBody').append(driver.deliveryDriverIsLate());
12             $('#deliverToast').toast('show');
13             setTimeout(() => {
14                 $('#deliverToastBody').empty()
15             }, 10000);
16         }
17     }
18
19     let drivers = [];
20
21     function addDelivery() {
22         var iconDeliver = $('select[name="icon"]').val();
23         var nameDeliver = $('input[name="name"]').val();
24         var surnameDeliver = $('input[name="surname"]').val();
25         var numberDeliver = $('input[name="number"]').val();
26         var addressDeliver = $('input[name="adresse"]').val();
27         var returnDeliver = $('input[name="arrivee"]').val();
28
29         var tableRowDeliver = '<tr><td>' + iconDeliver + '</td><td>' + nameDeliver + '</td><td>' + surnameDeliver + '</td><td>' + numberDeliver + '</td><td>' + addressDeliver + '</td><td>' + returnDeliver + '</td></tr>';
30
31         //validates the different criterias
32         function validateDelivery() {
33             if (isNaN(numberDeliver) || numberDeliver == "") {
34                 alert("No number provided. Try again please");
35                 return false;
36             } else if (numberDeliver.length != 10) {
37                 alert("The phonenumber needs to contain 8 digits");
38                 return false;
39             } else if (nameDeliver == "" || surnameDeliver == "" || addressDeliver == "" || returnDeliver == "") {
40                 alert("need valid input");
41                 return false;
42             } else {
43                 $(tableRowDeliver).appendTo('#driver_table');
44                 return true;
45             }
46         }
47
48         // validates and adds delivery driver
49         if (validateDelivery()) {
50             //highlight row
51             $('#driver_table tr').on('click', function () {
52                 $('table tr').removeClass('highlight_driver');
53                 $(this).addClass('highlight_driver');
54             });
55
56             //makes an class object delivery driver
57             let driver = new Delivery_driver(nameDeliver, surnameDeliver, iconDeliver, numberDeliver, addressDeliver, returnDeliver);
58             drivers.push(driver);
59             console.log(drivers)
60
61             //toast-delivery
62             deliveryDriverIsLate(driver.id);
63
64             //clears the input field after done
65             $("input").val("");
66         }
67     }
68
69     /*----- clear button removes delivery ----- */
70     function deliveryclear() {
71         if (confirm("Are you sure you want to remove this delivery?")) {
72             $('table tr').empty();
73         } else {
74             $('tbody > tr').removeClass('highlight_driver');
75         }
76     }
77 };
```

Figure 4 Delivery js file

StaffTabell for API Integration

“StaffTabell” js file defines an asynchronous function “staffUser()” that fetches data from the “randomuser.me”’s API to retrieve details of five staff members. It creates instances of the “Staff_Member” class using the fetched data and populates an HTML table with staff information like name, email, and images.

Additionally, it includes functionalities for:

- Highlighting rows in the table upon click.
- Converting time into the correct format.
- Handling staff “out” actions, setting out times, expected return times, and durations.
- Managing staff “in” actions to clear cells and rest staff status.
- Generating toast notifications for late staff members based on their expected return time.

See figure 5 down (to mention so the figure is many figures or whole code inside the “staffTable.js”

```
1  /*----- get random users -----*/
2  async function staffUserGet() {
3    const response = await fetch('https://randomuser.me/api/?results=5&inc-picture,name,email');
4    const data = await response.json();
5
6    const user = data.results[0];
7    const user1 = data.results[1];
8    const user2 = data.results[2];
9    const user3 = data.results[3];
10   const user4 = data.results[4];
11
12   /*----- staff members objects -----*/
13   let staff_0 = new Staff_Member(`${user.name.first}`, `${user.name.last}`, `${user.picture}`, `${user.email}`);
14   let staff_1 = new Staff_Member(`${user1.name.first}`, `${user1.name.last}`, `${user1.picture}`, `${user1.email}`);
15   let staff_2 = new Staff_Member(`${user2.name.first}`, `${user2.name.last}`, `${user2.picture}`, `${user2.email}`);
16   let staff_3 = new Staff_Member(`${user3.name.first}`, `${user3.name.last}`, `${user3.picture}`, `${user3.email}`);
17   let staff_4 = new Staff_Member(`${user4.name.first}`, `${user4.name.last}`, `${user4.picture}`, `${user4.email}`);
18
19   console.log(staff_0);
20   console.log(staff_1);
21   console.log(staff_2);
22   console.log(staff_3);
23   console.log(staff_4);
24
25   //populate table with staff members objects
26   const firstname = document.getElementById('firstname_0');
27   const lastname = document.getElementById('lastname_0');
28   const email = document.getElementById('email_0');
29   const img = document.getElementById('img_0');
30
31   firstname.innerText = staff_0.name;
32   lastname.innerText = staff_0.surname;
33   email.innerText = staff_0.email;
34   img.setAttribute('src', staff_0.picture);
35
36   const firstname1 = document.getElementById('firstname_1');
37   const lastname1 = document.getElementById('lastname_1');
38   const email1 = document.getElementById('email_1');
39   const img1 = document.getElementById('img_1');
40
41   firstname1.innerText = staff_1.name;
42   lastname1.innerText = staff_1.surname;
43   email1.innerText = staff_1.email;
44   img1.setAttribute('src', staff_1.picture);
45
46   const firstname2 = document.getElementById('firstname_2');
47   const lastname2 = document.getElementById('lastname_2');
48   const email2 = document.getElementById('email_2');
49   const img2 = document.getElementById('img_2');
50
51   firstname2.innerText = staff_2.name;
52   lastname2.innerText = staff_2.surname;
53   email2.innerText = staff_2.email;
54   img2.setAttribute('src', staff_2.picture);
55
56   const firstname3 = document.getElementById('firstname_3');
57   const lastname3 = document.getElementById('lastname_3');
58   const email3 = document.getElementById('email_3');
59   const img3 = document.getElementById('img_3');
60
61   firstname3.innerText = staff_3.name;
62   lastname3.innerText = staff_3.surname;
63   email3.innerText = staff_3.email;
64   img3.setAttribute('src', staff_3.picture);
65
66   const firstname4 = document.getElementById('firstname_4');
67   const lastname4 = document.getElementById('lastname_4');
68   const email4 = document.getElementById('email_4');
69   const img4 = document.getElementById('img_4');
70
71   firstname4.innerText = staff_4.name;
72   lastname4.innerText = staff_4.surname;
73   email4.innerText = staff_4.email;
74   img4.setAttribute('src', staff_4.picture);
75
76   /*----- staff table -----*/
77   //highlight row on click
78   $(function () {
79     $('#person-table tr').on('click', function () {
80       $('#table tr').removeClass('highlight_row');
81       $(this).addClass('highlight_row');
82     });
83   });
84
85   //converts time into right format
86   const time_convert = (time) => {
87     let h = Math.floor(time / 60);
88     let m = time % 60;
89     (h < 10) ? h = '0' + h : h;
90     (m < 10) ? m = '0' + m : m;
91     return `${h}:${m}`;
92   };
93
94   /*----- staff out button -----*/
95   function staffOut() {
96     $('#staff_out').on('click', function () {
97       var time = prompt('Input time in minutes:');
98       if (time === null) return;
99
100      //out time
101      let tid = new Date(new Date().getTime()).toLocaleTimeString();
102
103      if ($('#row_1').hasClass('highlight_row')) {
104        $('#outtime_0').empty();
105        staff_0.outtime = tid;
106        $('#outtime_0').append(staff_0.outtime);
107      } else if ($('#row_1').hasClass('highlight_row')) {
108        $('#outtime_1').empty();
109        staff_1.outtime = tid;
110        $('#outtime_1').append(staff_1.outtime);
111      } else if ($('#row_2').hasClass('highlight_row')) {
112        $('#outtime_2').empty();
113        staff_2.outtime = tid;
114        $('#outtime_2').append(staff_2.outtime);
115      } else if ($('#row_3').hasClass('highlight_row')) {
116        $('#outtime_3').empty();
117      }
118    });
119  }
120
121  } else {
122    //duration
123    if ($('#row_0').hasClass('highlight_row')) {
124      $('#timeaway_0').empty();
125      staff_0.duration = time_convert(time);
126      $('#timeaway_0').append(staff_0.duration);
127    } else if ($('#row_1').hasClass('highlight_row')) {
128      $('#timeaway_1').empty();
129      staff_1.duration = time_convert(time);
130      $('#timeaway_1').append(staff_1.duration);
131    } else if ($('#row_2').hasClass('highlight_row')) {
132      $('#timeaway_2').empty();
133      staff_2.duration = time_convert(time);
134      $('#timeaway_2').append(staff_2.duration);
135    } else if ($('#row_3').hasClass('highlight_row')) {
136      $('#timeaway_3').empty();
137      staff_3.duration = time_convert(time);
138      $('#timeaway_3').append(staff_3.duration);
139    } else if ($('#row_4').hasClass('highlight_row')) {
140      $('#timeaway_4').empty();
141      staff_4.duration = time_convert(time);
142      $('#timeaway_4').append(staff_4.duration);
143    }
144  }
145
146  }
147
148  }
149
150  }
151
152  }
153
154  }
155
156  }
157
158  }
159
160  }
161
162  }
163
164  }
165
166  }
167
168  }
169
170  }
171
172  }
173
174  }
175
176  }
177
178  }
179
180  }
181
182  }
183
184  }
185
186  }
187
188  }
189
190  }
191
192  }
193
194  }
195
196  }
197
198  }
199
200  }
201
202  }
203
204  }
205
206  }
207
208  }
209
210  }
211
212  }
213
214  }
215
216  }
217
218  }
219
220  }
221
222  }
223
224  }
225
226  }
227
228  }
229
230  }
231
232  }
233
234  }
235
236  }
237
238  }
239
240  }
241
242  }
243
244  }
245
246  }
247
248  }
249
250  }
251
252  }
253
254  }
255
256  }
257
258  }
259
260  }
261
262  }
263
264  }
265
266  }
267
268  }
269
270  }
271
272  }
273
274  }
275
276  }
277
278  }
279
280  }
281
282  }
283
284  }
285
286  }
287
288  }
289
290  }
291
292  }
293
294  }
295
296  }
297
298  }
299
300  }
301
302  }
303
304  }
305
306  }
307
308  }
309
310  }
311
312  }
313
314  }
315
316  }
317
318  }
319
320  }
321
322  }
323
324  }
325
326  }
327
328  }
329
330  }
331
332  }
333
334  }
335
336  }
337
338  }
339
340  }
341
342  }
343
344  }
345
346  }
347
348  }
349
350  }
351
352  }
353
354  }
355
356  }
357
358  }
359
360  }
361
362  }
363
364  }
365
366  }
367
368  }
369
370  }
371
372  }
373
374  }
375
376  }
377
378  }
379
380  }
381
382  }
383
384  }
385
386  }
387
388  }
389
390  }
391
392  }
393
394  }
395
396  }
397
398  }
399
400  }
401
402  }
403
404  }
405
406  }
407
408  }
409
410  }
411
412  }
413
414  }
415
416  }
417
418  }
419
420  }
421
422  }
423
424  }
425
426  }
427
428  }
429
430  }
431
432  }
433
434  }
435
436  }
437
438  }
439
440  }
441
442  }
443
444  }
445
446  }
447
448  }
449
450  }
451
452  }
453
454  }
455
456  }
457
458  }
459
460  }
461
462  }
463
464  }
465
466  }
467
468  }
469
470  }
471
472  }
473
474  }
475
476  }
477
478  }
479
480  }
481
482  }
483
484  }
485
486  }
487
488  }
489
490  }
491
492  }
493
494  }
495
496  }
497
498  }
499
500  }
501
502  }
503
504  }
505
506  }
507
508  }
509
510  }
511
512  }
513
514  }
515
516  }
517
518  }
519
520  }
521
522  }
523
524  }
525
526  }
527
528  }
529
530  }
531
532  }
533
534  }
535
536  }
537
538  }
539
540  }
541
542  }
543
544  }
545
546  }
547
548  }
549
550  }
551
552  }
553
554  }
555
556  }
557
558  }
559
560  }
561
562  }
563
564  }
565
566  }
567
568  }
569
570  }
571
572  }
573
574  }
575
576  }
577
578  }
579
580  }
581
582  }
583
584  }
585
586  }
587
588  }
589
590  }
591
592  }
593
594  }
595
596  }
597
598  }
599
600  }
601
602  }
603
604  }
605
606  }
607
608  }
609
610  }
611
612  }
613
614  }
615
616  }
617
618  }
619
620  }
621
622  }
623
624  }
625
626  }
627
628  }
629
630  }
631
632  }
633
634  }
635
636  }
637
638  }
639
640  }
641
642  }
643
644  }
645
646  }
647
648  }
649
650  }
651
652  }
653
654  }
655
656  }
657
658  }
659
660  }
661
662  }
663
664  }
665
666  }
667
668  }
669
670  }
671
672  }
673
674  }
675
676  }
677
678  }
679
680  }
681
682  }
683
684  }
685
686  }
687
688  }
689
690  }
691
692  }
693
694  }
695
696  }
697
698  }
699
700  }
701
702  }
703
704  }
705
706  }
707
708  }
709
710  }
711
712  }
713
714  }
715
716  }
717
718  }
719
720  }
721
722  }
723
724  }
725
726  }
727
728  }
729
730  }
731
732  }
733
734  }
735
736  }
737
738  }
739
740  }
741
742  }
743
744  }
745
746  }
747
748  }
749
750  }
751
752  }
753
754  }
755
756  }
757
758  }
759
760  }
761
762  }
763
764  }
765
766  }
767
768  }
769
770  }
771
772  }
773
774  }
775
776  }
777
778  }
779
780  }
781
782  }
783
784  }
785
786  }
787
788  }
789
790  }
791
792  }
793
794  }
795
796  }
797
798  }
799
800  }
801
802  }
803
804  }
805
806  }
807
808  }
809
810  }
811
812  }
813
814  }
815
816  }
817
818  }
819
820  }
821
822  }
823
824  }
825
826  }
827
828  }
829
830  }
831
832  }
833
834  }
835
836  }
837
838  }
839
840  }
841
842  }
843
844  }
845
846  }
847
848  }
849
850  }
851
852  }
853
854  }
855
856  }
857
858  }
859
860  }
861
862  }
863
864  }
865
866  }
867
868  }
869
870  }
871
872  }
873
874  }
875
876  }
877
878  }
879
880  }
881
882  }
883
884  }
885
886  }
887
888  }
889
890  }
891
892  }
893
894  }
895
896  }
897
898  }
899
900  }
901
902  }
903
904  }
905
906  }
907
908  }
909
910  }
911
912  }
913
914  }
915
916  }
917
918  }
919
920  }
921
922  }
923
924  }
925
926  }
927
928  }
929
930  }
931
932  }
933
934  }
935
936  }
937
938  }
939
940  }
941
942  }
943
944  }
945
946  }
947
948  }
949
950  }
951
952  }
953
954  }
955
956  }
957
958  }
959
960  }
961
962  }
963
964  }
965
966  }
967
968  }
969
970  }
971
972  }
973
974  }
975
976  }
977
978  }
979
980  }
981
982  }
983
984  }
985
986  }
987
988  }
989
990  }
991
992  }
993
994  }
995
996  }
997
998  }
999
1000 }
```

```
145     $('#routeTime_1').empty();
146     staff_1.outTime = tid;
147     $('#routeTime_1').append(staff_1.outTime);
148     } else if ($('#row_2').hasClass('highlight_row')) {
149     $('#routeTime_2').empty();
150     staff_2.outTime = tid;
151     $('#routeTime_2').append(staff_2.outTime);
152     } else if ($('#row_3').hasClass('highlight_row')) {
153     $('#routeTime_3').empty();
154     staff_3.outTime = tid;
155     $('#routeTime_3').append(staff_3.outTime);
156     } else {
157     $('#routeTime_4').empty();
158     staff_4.outTime = tid;
159     $('#routeTime_4').append(staff_4.outTime);
160     }
161 }
162 //return time
163 let returnTime = new Date(new Date().getTime() + time * 60000).toLocaleTimeString();
164 if ($('#row_0').hasClass('highlight_row')) {
165     $('#expectedReturn_0').empty();
166     staff_0.expectedReturnTime = returnTime;
167     $('#expectedReturn_0').append(staff_0.expectedReturnTime);
168     } else if ($('#row_1').hasClass('highlight_row')) {
169     $('#expectedReturn_1').empty();
170     staff_1.expectedReturnTime = returnTime;
171     $('#expectedReturn_1').append(staff_1.expectedReturnTime);
172     } else if ($('#row_2').hasClass('highlight_row')) {
173     $('#expectedReturn_2').empty();
174     staff_2.expectedReturnTime = returnTime;
175     $('#expectedReturn_2').append(staff_2.expectedReturnTime);
176     } else if ($('#row_3').hasClass('highlight_row')) {
177     $('#expectedReturn_3').empty();
178     staff_3.expectedReturnTime = returnTime;
179     $('#expectedReturn_3').append(staff_3.expectedReturnTime);
180     } else {
181     $('#expectedReturn_4').empty();
182     staff_4.expectedReturnTime = returnTime;
183     $('#expectedReturn_4').append(staff_4.expectedReturnTime);
184     }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
```

Figure 5 Staff Table.js

HTML File

The HTML serves as a dashboard for managing staff members, deliveries, and schedules see figure 6 down below (contain the first page of the Html code and the rest of you can check when you open the web application). It includes several sections:

- Navbar: Contains navigation links for Dashboard, inventory, and orders, with dropdown menus for specific actions.
- Header Section: displays the logo and a title for the reception management dashboard.
- Staff table section: show a table of staff members with details like name, surname, email, status, clock-in/out times, and expected return times. Buttons for clocking in and out are included.
- Toast notifications: alerts for late staff members, triggered with relevant information about the latte employee.
- Schedule Delivery Section: enables scheduling deliveries with fields for vehicle type, recipient's name, surname, telephone, delivery address, and return time.
- Delivery Board Section: Displays a table for delivery drivers with columns for vehicle, name, surname, telephone, delivery address, and return time. Also includes a button to clear deliveries.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
7   <!--motorcycle icon-->
8   <script src="https://kit.fontawesome.com/b57ae05b39.js" crossorigin="anonymous"></script>
9   <script src="https://cdn.jsdelivr.net/npm/popper.js/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVubq5LC7Qb9DXgDA9i+tQ8Zj31wAwPtgtFxb18NT4GN1R8p" crossorigin="anon"
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhWJC2AL4u+LWgxfKTRICfue0JTtR+EQDz/bgldoEyl4HbzUFP0Kbr30EcQF" crossorigin="anonymo"
11  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" integrity="sha384-cVKIPhWJC2AL4u+LWgxfKTRICfue0JTtR+EQDz/bgldoEyl4HbzUFP0Kbr30EcQF" crossorigin="anonymo"
12  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Consolas">
13  <script src="https://jsdelivr.net/npm/axios@0.21.0/dist/axios.min.js"></script>
14  <script src="https://jsdelivr.net/npm/axios@0.21.0/dist/axios.min.js"></script>
15  <script src="https://jsdelivr.net/npm/axios@0.21.0/dist/axios.min.js"></script>
16  <script src="https://jsdelivr.net/npm/axios@0.21.0/dist/axios.min.js"></script>
17  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-icons/1.9.1/font/bootstrap-icons.css">
18  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/5.1.0/css/bootstrap.min.css">
19
20 <title>Document</title>
21 </head>
22 <body>
23   <!--navbar-->
24   <nav class="navbar nav-justified navbar-expand-sm navbar-dark">
25     <div class="container-fluid">
26
27       <!--this makes a hamburger menu when frame is sm-->
28       <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
29         data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
30         aria-label="Toggle navigation">
31         <span class="navbar-toggler-icon"></span>
32       </button>
33
34       <div class="collapse navbar-collapse" id="navbarSupportedContent">
35         <ul class="navbar-nav">
36           <li class="nav-item">
37             <a class="nav-link active" aria-current="page" href="#">Dashboard</a>
38           </li>
39           <li class="nav-item dropdown">
40             <a class="nav-link active dropdown-toggle" href="#" id="navbarDropdown"
41               data-bs-toggle="dropdown" aria-expanded="false">
42               Inventory
43             </a>
44           </li>
45         </ul>
46       </div>
47     </div>
48   </nav>
49 </body>
50 </html>
```

Figure 6 HTML

Challenges Faced and Solutions Implemented

During the project, I encountered several challenges that required troubleshooting and innovative solutions. These challenges included difficulties in populating tables with data, issues with API requests due to missing file linking, malfunctioning time formatting, and problems with toast notifications.

The resolution process involved meticulous debugging, such as rectifying file linking errors, restructuring code classes to streamline data population, and revising the toast notification system. Utilizing relative positioning and changing its trigger to a button significantly contributed to resolving the toast display issue. Additionally, optimizing the design in Bootstrap required numerous iterations and class modifications to achieve the desired layout and functionality.

Overall, the project entailed comprehensive development and problem-solving, resulting in a robust system for staff management and time-related operations.

This endeavor provided invaluable experience in software development, debugging, and the utilization of various technologies to create an efficient and user-friendly system.

-----the end -----