



Inspiring Excellence

CSE471: System Analysis and Design
Project Report
Project Title: Movie Review

Group No:03, CSE471 Lab Section: 08, Summer2023		
ID	Name	Contribution
21101010	Kamran Hassan Shomrat	Backend-MERN
21101163	Mamsad Ibn Yeahia	MERN – Search, Profile
21101169	Nafiul Majid	MERN - Post, API

Table of Contents

Section No	Content	Page No
1	Introduction	3
2	Functional Requirements	3
3	User Manual	4
4	Frontend Development	7
5	Backend Development	10
6	Technology (Framework, Languages)	13
7	Github Repo Link	13
8	Individual Contribution	13-14

Introduction

Welcome to our latest project, where the captivating world of cinema meets the power of modern technology: the "Movie Review System - using MERN." In a time when movies have become an integral part of our culture, influencing our thoughts, emotions, and conversations, we present to you a sophisticated platform that not only connects movie enthusiasts but also empowers them to express, explore, and engage with films on a whole new level. Harnessing the prowess of the MERN (MongoDB, Express.js, React, Node.js) stack, our project aims to revolutionize the way we interact with movies. With an ever-expanding universe of films and television shows, it's often challenging to navigate and decide what to watch next. This is where our Movie Review System comes into play – offering a seamless and dynamic solution for users to find, review, and discuss their favorite (or not-so-favorite) movies. Our system not only provides a user-friendly interface for browsing an extensive collection of movies but also incorporates real-time reviews and ratings from fellow users. This ensures that every movie-goer's voice is heard, helping others make informed decisions about what to watch. Through a meticulously crafted React frontend, backed by a robust Express.js and Node.js backend, we ensure a smooth and responsive experience for users across various devices.

Functional Requirements

1. User Authentication and Authorization:

The system shall support user registration and login using secure authentication methods. Users shall be categorized into roles such as regular users and administrators.

2. Movie Database Integration:

The server shall integrate with a comprehensive movie database to fetch movie details including titles, genres, release dates, and descriptions. Movie information fetched from the external database shall be stored in the system's local database for efficient retrieval.

3. User Reviews and Ratings:

Registered users shall be able to submit reviews and ratings for movies. Users shall have the ability to edit and delete their own reviews. The system shall calculate and display average ratings for each movie based on user submissions.

4. User Interactions:

Users shall have the option to like, dislike, or comment on reviews submitted by other users. Users shall receive notifications for interactions on their reviews, comments, and other relevant activities.

5. User Profile Management:

Users shall have individual profiles to manage their account information, including profile pictures and personal details. Profiles shall display users' reviewed movies, ratings, and interactions.

6. API Documentation:

The server shall provide comprehensive API documentation for developers to interact with the backend services. The documentation shall include endpoints, request methods, parameters, and responses.

7. Security Measures:

The server shall implement secure communication using encryption protocols (JWT). User inputs, such as reviews and comments, shall be sanitized to prevent security vulnerabilities like cross-site scripting (XSS) attacks.

8. Admin Dashboard:

Administrators shall have access to a dedicated dashboard for managing user accounts, reported content, and system settings.

By fulfilling these functional requirements, the "Movie Review Server - using MERN" project will deliver a robust, user-friendly, and feature-rich platform for movie enthusiasts to interact, review, and engage with films in an innovative way.

Frontend Development

Frontend Development: The process of creating HTML, CSS, and JavaScript for a website or Web application so that a user can see and interact with them directly is known as front-end web development, sometimes known as client-side development. The difficulty with front end development is that the methods and technologies used to build a website's front end are continually changing, necessitating constant monitoring of the industry's advancements. The goal of website design is to present the content in an understandable and pertinent manner to users as soon as they access the site. This is made much more difficult by the fact that customers today utilize a wide range of devices with different screen sizes and resolutions, prompting the site's designer to take these factors into account. They must make sure that their website functions properly across a variety of browsers (cross-browser), operating systems (cross-platform), and devices (cross-device), which necessitates considerable preparation on the part of the developer.

Contribution of: Kamran Hassan Shomrat, Nafiul Majid, Mamsad Ibn Yeahia

.Contribution of ID: 21101169, Name: Nafiul Majid

Set up the Background, Color and Interface of admin and user login.

#Admin login

```
src > components > admin > JS Admin.js > [0] Admin
16   }
17   const useStyles = makeStyles((theme) =>({
18
19     grid:{
20       marginTop:theme.spacing(9),
21       height:900,
22       display:'flex',
23       flexDirection: 'column',
24       textAlign:'center',
25       backgroundImage:`url("${https://cutewallpaper.org/21/admin-wallpaper/Admin-Wallpapers-Desktop-Wallpaper.jp
26       justifyContent:'center',
27       backdropFilter: "blur(5px)",
28     },
29     paperStyle:{
30
31       padding :20,
32       height:'70vh',
33       width:500,
34       margin:"20px auto",
35
36     },
37     password:{
38       marginTop: theme.spacing(2)
39     },
40     color:{
41       color:"blue"
42     },
43     admin:{
```

#User Login

```
26
27   const useStyles = makeStyles((theme) =>({
28
29     grid:{
30       marginTop:theme.spacing(10),
31       height:'100%',
32       display:'flex',
33       flexDirection: 'column',
34       textAlign:'center',
35     },
36     snackbar:{
37       width: '100%',
38       '& > * + *': {
39         marginTop: theme.spacing(2),
40       },
41     },
42     password:{
43       marginTop: theme.spacing(2)
44     },
45     signup:{
46       marginTop: theme.spacing(2)
47     },
48     paperStyle:{
49       padding :20,
50       height:'70vh',
51       width:500,
52       margin:"20px auto",
53     },
```

Contribution of ID: 21101169, Name: Nafiul Majid

Set up the Background, Color and Interface of the Navbar and Homepage.

#Navbar

```
src > components > ... Navbar.jsx > ...
17   const useStyles = makeStyles((theme) =>({
18     Toolbar:{
19
20       display:"flex",
21       justifyContent:"space-between",
22
23     },
24   },
25   logoLg:{
26
27     display : "block",
28     marginTop:theme.spacing(1),
29     [theme.breakpoints.up("sm")]:{
30       display: "block",
31     },
32   },
33   logoSm:{
34
35     display : "block",
36     marginTop:theme.spacing(1),
37     [theme.breakpoints.up("md")]:{
38       display: "block",
39     },
40   },
41   search:{
42     marginLeft: theme.spacing(1),
43     display: "flex",
44     alignItems: "center",
45     position: 'relative',
46     borderRadius: theme.shape.borderRadius,
47     backgroundColor: alpha(theme.palette.co
48     '&:hover': {
49       backgroundColor: alpha(theme.palette.co
50   },
51
52   src > components > ... Navbar.jsx > ...
53   },
54   InputBase:{
55     marginLeft: theme.spacing(1),
56     color: "white",
57     marginLeft: theme.spacing(1),
58     width: "100%"
59   },
60   },
61   icons:{
62     display: "flex",
63     alignItems: "center",
64     marginLeft: theme.spacing(1),
65     [theme.breakpoints.down("sm")]:{
66       display: (props) => (props.open ? "
67   },
68   },
69   badge:{
70     marginRight: theme.spacing(2)
71   },
72   },
73   searchButton:{
74     marginRight: theme.spacing(1),
75     [theme.breakpoints.up("sm")]:{
76       display: "none"
77   },
78   },
79   },
80   cancel:{
81     display:"none",
82     [theme.breakpoints.down("sm")]:{
83       display: "flex"
84   },
85   },
86   button:{
87     display: "flex",
88     align-items: "center",
89     justify-content: "center",
90     width: "100%",
91     height: "100%",
92     border: "1px solid #ccc",
93     border-radius: "50%",
94     background-color: "white",
95     color: "black",
96     text-align: "center",
97     line-height: "1",
98     padding: "5px 10px",
99     cursor: "pointer",
100    transition: "background-color 0.3s, color 0.3s",
101    '&:hover': {
102      background-color: "black",
103      color: "white",
104    },
105    '&:disabled': {
106      background-color: "white",
107      color: "black",
108      cursor: "not-allowed",
109    },
110  },
111 },
112 },
113 },
114 },
115 },
116 },
117 },
118 },
119 },
120 },
121 },
122 },
123 },
124 },
125 },
126 },
127 },
128 },
129 },
130 },
131 },
132 },
133 },
134 },
135 },
136 },
137 },
138 },
139 },
140 },
141 },
142 },
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 },
76
```

#Homepage

```

src > components > tmdb_postjsx > ...
13  const useStyles = makeStyles(({theme}) =>({
14      container:{
15
16          marginBottom:theme.spacing(5),
17          marginTop:theme.spacing(2),
18          height:'100%',
19          display:'flex',
20          flexDirection: 'column'
21      },
22      media:{
23          height:350,
24          paddingTop: '56.25%',
25          [theme.breakpoints.down("sm")]:{
26              height:350
27          }
28      },
29      setup:{
30          marginTop:theme.spacing(5),
31      },
32      numberOfLines:{
33          WebkitLineClamp:1
34      },
35      pagination:{
36          marginTop: theme.spacing(5),
37          display: "flex",
38          justifyContent: "center"
39      },
40      root: {
41          borderRadius:15,
42          background: 'white',
43          marginRight: theme.spacing(2)
44      },
45
src > components > tmdb_postjsx > ...
47      textField: {
48          marginLeft: theme.spacing(0),
49          marginRight: theme.spacing(2),
50          width: '43vh',
51          marginTop: theme.spacing(12),
52          justifyContent: "center",
53          [theme.breakpoints.up("sm")]:{
54              marginLeft: theme.spacing(8),
55          }
56      },
57
58  },
59  loading: {
60
61      width: '100%',
62      justifyContent:'center',
63      '& > * + *': {
64          marginTop: theme.spacing(2),
65      },
66
67  },
68
69
70  box:{
71      display:'flex',
72      flexDirection: 'column',
73      marginTop: theme.spacing(1),
74      height:"100vh",
75      marginLeft:theme.spacing(4),
76      marginRight:theme.spacing(4),
77      marginTop:theme.spacing(4)
78  },
79

```

Contribution of ID: 21101163, Name: Mamsad Ibn Yeahia

Set up the Background, Color and Interface of Profile and Search

#Profile

```
src > components > Profile > profiles.jsx > ...
49 const useStyles = makeStyles((theme) =>({
50   root: {
51     width: '100%',
52     maxWidth: 800,
53     backgroundColor: theme.palette.background.paper,
54     display: 'flex',
55     marginLeft: theme.spacing(5),
56     [theme.breakpoints.down("xs")]: {
57     },
58   },
59   },
60   root2: {
61     width: '100%',
62     maxWidth: 1050,
63     backgroundColor: theme.palette.background.paper,
64     display: 'flex',
65     justifyContent: 'space-between',
66     marginLeft: theme.spacing(5),
67     [theme.breakpoints.down("xs")]: {
68     flexDirection: 'column',
69   },
70   },
71   },
72   container: {
73     marginBottom: theme.spacing(5),
74     marginTop: theme.spacing(2),
75     height: '100%',
76     display: 'flex',
77   },
78   media: {
79     height: 100,
80     paddingTop: '56.25%',
81     [theme.breakpoints.down("sm")]: {
82     height: 350,
83   },
84   },
85   containerNEW: {
86     marginBottom: theme.spacing(5),
87     marginTop: theme.spacing(2),
88     height: '100%',
89     display: 'flex',
90     flexDirection: 'column'
91   },
92   },
93   Media: {
94     height: 350,
95     paddingTop: '56.25%',
96     [theme.breakpoints.down("sm")]: {
97     height: 350
98   },
99   },
100   setup: {
101     marginTop: theme.spacing(5),
102   },
103   numberOfLines: {
104     WebkitLineClamp: 1
105   },
106   purple: {
107     color: theme.palette.getContrastText(deepPurple),
108     backgroundColor: deepPurple[500],
109     marginLeft: theme.spacing(3)
110   },
111   watchlist: {
112     marginTop: theme.spacing(2)
113   },
114   icon: {
115     minWidth: '100%',
116     display: "flex",
117   },
118   },
119   },
120   },
121   },
122   },
123   },
124   },
125   },
126   },
127   },
128   },
129   },
130   },
131   },
132   },
133   },
134   },
135   },
136   },
137   },
138   },
139   },
140   },
141   },
142   },
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  }
```

#Search

```

src > components > search > Dbsearch.jsx > ...
10 const useStyles = makeStyles(({theme}) =>({
11
12   container:{
13     marginBottom:theme.spacing(5),
14     marginTop:theme.spacing(2),
15     height:'100%',
16     display:'flex',
17     flexDirection: 'column'
18   },
19
20   media:{
21     height:350,
22     padding: '56.25%',
23     [theme.breakpoints.down("sm")]:{
24       height:350
25     }
26   },
27   setup:{
28     marginTop:theme.spacing(5),
29   },
30   numberOfLines:{
31     WebkitLineClamp:1
32   },
33   pagination:{
34     marginTop: theme.spacing(5),
35     display: "flex",
36     justifyContent: "center"
37   },
38   root: {
39     borderRadius:15,
40     background: 'white',
41     position: '-webkit-sticky',
42     position: 'sticky',
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78 const Search = () => {
79
80   let baseUrl;
81   const classes = useStyles();
82   const [trend, setTrend] = useState([]);
83   const [search,setSearch] =useState();
84
85   const [loading,setLoading] =useState(false);
86   const [currentPage, setCurrentPage] = useState(1);
87   const [postsPerPage,setPostsPerPage] = useState(5);
88
89   const history = useHistory();
90
91   const handleRoute = (result, type) =>{
92
93     history.push('details/'+result+'/'+type)
94   }
95
96   const getRequest = async()=>{
97
98
99
100     setLoading(true);
101
102
103     (search ==null || search == "")? baseUrl:
104
105     const get = await axios.get(baseUrl, {
106       params:{
107         api_key:"c92ecd56753067461e71f400f3;
108         language:"en-US",
109         page:1,
110         include_adult:false,

```

```

    )))
const Admin=()=>{
  const [anchorEl, setAnchorEl] = React.useState(null);
  const [email, setEmail] = useState("");
  const [password,setPassword] = useState("");
  const [check,setCheck] =useState(false);
  const [open,setOpen] =useState({
    status:false,
    message: " ",
  })

  const history = useHistory();
  const {state} = useLocation();
  const classes = useStyles();

  const handleRoute = (result) =>{

    history.push('/'+result)
  }

```



```

    },config);
    console.log(data.data);
    localStorage.setItem("authToken", data.data.token);
    console.log(localStorage.getItem("authToken"));
    config = {
      headers:{
        "content-Type":"application/json",
        "x-auth-token": localStorage.getItem("authToken")
      }
    }

    data = await axios.get(host+"/api/auth",config);
    if(data.data.email == "admin@gmail.com"){
      if(data.data){
        setOpen({
          status:true,
          message: "Login Successful",
          severity: "success",
        },
        setCheck(true))
        history.push(state?.from || '/')
        window.location.reload(false);
      }
    }
  }
}

```

Backend Development

Backend Development: In the development of the "Movie Review - using MERN" project's backend, our focus lies on creating a robust, efficient, and secure foundation that powers the entire cinematic experience. Leveraging the MERN (MongoDB, Express.js, React, Node.js) stack, our backend architecture ensures seamless communication with the frontend while enabling dynamic interactions and real-time data updates. At the core of our backend development is MongoDB, a powerful NoSQL database that stores movie details, user information, reviews, and interactions. This allows for flexible data modeling and rapid retrieval of information. Express.js serves as the middleware, facilitating RESTful APIs that connect the frontend and backend components. This enables the frontend to seamlessly retrieve and submit data, ensuring a smooth user experience. Node.js, with its event-driven architecture, powers our backend server, handling concurrent connections and delivering high performance. It serves as the bridge between the database and the frontend, managing requests, processing data, and orchestrating the flow of information. Our backend development team ensures data security by implementing encryption protocols and input

validation to prevent common vulnerabilities. User authentication and authorization mechanisms are carefully crafted to protect user data and manage access rights. To enable real-time interactions and updates, we incorporate websockets technology, allowing users to receive immediate notifications for comments, likes, and other interactions on their reviews. The backend development process also focuses on scalability and performance optimization. With carefully designed APIs, database indexing, and efficient query execution, we guarantee a responsive system that can handle increasing user engagement and growing movie databases. Regular testing, debugging, and performance monitoring are integral parts of our backend development workflow. This ensures a stable and reliable backend that complements the frontend's engaging user interface. As we collaborate with the frontend team to seamlessly integrate both components, the "Movie Review - using MERN" project's backend promises to deliver a cutting-edge movie review platform that thrives on modern technology and innovation.

Contribution of: Kamran Hassan Shomrat, Nafiul Majid, [MAMSAD IBN YEAHIA](#)

#Index.Jsx

```
const express = require('express');
const users = require('./Controllers/users');
const profiles = require('./Controllers/profiles');
const posts = require('./Controllers/posts');
const auth = require('./Controllers/auth');
const cors = require('cors');

const connectdb = require('./config/db');

const app = express();
app.use(cors());
app.use(express.json({ extended: false }));

connectdb();

app.get('/', (req, res) => {
  return res.send('app is running');
});

app.use('/api/auth', auth);
app.use('/api/profiles', profiles);
app.use('/api/users', users);
app.use('/api/posts', posts);

const PORT = process.env.PORT || 8000;

app.listen(PORT, () => {
```

#Schema:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
const postSchema = new Schema({
  user: {
    type : Schema.Types.ObjectId,
    ref: 'user'
  },
  text : {
    type: String,
    required: true
  },
  image : {
    type: String,
    required: true
  },
  shows : {
    type: String,
    required: true
  },
  name: {
    type: String,
  },
  avatar:{
    type:String,
  }
});
```

#Admin Login

```
import { Container, makeStyles, Grid } from "@material-ui/core";
import Post from "../post";
import { Tmdb_post } from "../tmdb_post";

const useStyles = makeStyles((theme) =>({
  container:{
    paddingTop:theme.spacing(10)
  }
})))
const AdminMain = () => {
  const classes = useStyles();

  return (
    <Container className={classes.container} cols={3}>
      <h1>Working</h1>
    </Container>
  )
};

export default AdminMain;
```

#User Login

```
const express = require('express');
const router = express.Router();
const gravatar = require('gravatar');

const bcrypt = require('bcrypt');

const User = require('../Models/User');
const jwt = require('jsonwebtoken');
const config = require('config');

const { check, validationResult } = require('express-validator');

router.get('/', (req, res) => {
  try{
    res.send('User users');
    res.status(200);
  }catch(error){
    console.log(error);
  }
});

router.get('/all' ,async(req, res) => {
  try {
    const user = await User.find();
    res.json(user)
  }
});
```

#Admin Panel

```
import React,{useState, useEffect} from 'react'
import { makeStyles, Grid,Paper, Avatar, TextField, Button, Typography,Link, responsiveFontSizes } from '@material-ui/core'

import axios from 'axios';
import { useHistory } from "react-router-dom";
import ExitToAppRoundedIcon from '@material-ui/icons/ExitToAppRounded';
import Snackbar from '@material-ui/core/Snackbar';
import MuiAlert from '@material-ui/lab/Alert';
import Table from '@material-ui/core/Table';
import TableBody from '@material-ui/core/TableBody';
import TableCell from '@material-ui/core/TableCell';
import TableContainer from '@material-ui/core/TableContainer';
import TableHead from '@material-ui/core/TableHead';
import TableRow from '@material-ui/core/TableRow';
import Select from '@material-ui/core/Select';
import {host} from '../host';

function Alert(props) {
  return <MuiAlert elevation={6} variant="filled" {...props} />;
}

const useStyles = makeStyles((theme) =>({
  grid:{
    marginTop:theme.spacing(10),
    height:'100%',
    display:flex;
```

#Post

```
Controllers > JS posts.js > router.delete('/:id') callback
53 router.post('/', [auth,
54   [
55     check('text', 'Text is required')
56     .not()
57     .isEmpty(),
58     check('image', 'image is required')
59     .not()
60     .isEmpty()
61   ]
62 ], async(req, res) => {
63   const errors = validationResult(req);
64   if(!errors.isEmpty()){
65     return res.status(400).json({errors: errors.array()});
66   }
67   try {
68     const user = await User.findById(req.user.id);
69     const post = new Post({
70       text : req.body.text,
71       image : req.body.image,
72       shows : req.body.shows,
73       name : user.name,
74       avatar: user.avatar,
75       user: req.user.id
76     });
77     await post.save();
78     res.json(post);
79   } catch (error) {
80     console.error(error.message);
81     res.status(500).send('Server Error');
82   }
83 })
84 //get post ByID
```

Framework

MERN. The MERN framework is a popular stack of technologies used for building web applications. It is an acronym that stands for MongoDB, Express.js, React, and Node.js. Each of these technologies plays a specific role in the development process, and when combined, they provide a powerful and efficient way to create dynamic and responsive web applications. MongoDB: MongoDB is a NoSQL database that stores data in a flexible, JSON-like format called BSON (Binary JSON). Express.js helps in handling routing, middleware creation, and various HTTP utility methods. React is a JavaScript library developed by Facebook for building user interfaces. Node.js is a runtime environment that allows developers to run JavaScript on the server side. Node.js enables developers to use JavaScript both on the frontend and backend, creating a more cohesive development experience. The MERN stack's popularity comes from its flexibility, performance, and the ability to build feature-rich web applications using JavaScript throughout the entire stack. However, it's worth noting that the MERN stack is just one of many options available for web development, and the choice of technology stack should be based on the specific requirements and goals of your project.

Github Repo Link

Link: <https://github.com/Pepsi101/Movie-Review-Server.git>

Individual Contribution

Mamsad's Contribution:

Throughout the development of the "Movie Review Server - using MERN," Mamsad's expertise and dedication were instrumental in shaping the project's frontend. His proficiency in React.js brought life to the APIs that connect our frontend components. Mamsad was pivotal in designing and implementing the user authentication and authorization mechanisms, ensuring secure user interactions and data privacy. His attention to detail and commitment to best practices resulted in efficient data processing and seamless API endpoints. Mamsad's collaborative spirit and proactive problem-solving approach significantly contributed to the project's success.

Kamran's Contribution:

Kamran's role in the "Movie Review Server - using MERN" project was pivotal in ensuring the integration of MongoDB as the database backbone. His expertise in database design and management facilitated the storage and retrieval of movie details, user profiles, and reviews. Kamran's dedication to optimizing database queries and structuring efficient data models led to a responsive system capable of handling a vast collection of movies and user interactions. His collaboration with frontend developers ensured smooth communication between the two components, delivering a cohesive user experience. Kamran's meticulous approach to database security and data validation added a layer of robustness to the backend infrastructure.

Nafiul's Contribution:

Nafiul's contribution to the "Movie Review Server - using MERN" project was focused on ensuring real-time interactions and dynamic updates. His expertise in WebSocket technology enabled the implementation of immediate notifications for user interactions, adding an engaging layer to the platform. Nafiul's skillful coding and integration of WebSocket libraries ensured that users receive prompt updates for comments, likes, and interactions on their reviews. His ability to work collaboratively, bridging the backend and frontend functionalities, made the project's real-time features a seamless reality. Nafiul's dedication to optimizing performance and ensuring a scalable solution enriched the project's overall success.

Together, the combined efforts of Mamsad, Kamran, and Nafiul in the "Movie Review Server - using MERN" project's backend development have resulted in a feature-rich, secure, and efficient platform that redefines the way users interact with movie reviews and engage with cinematic content.