



## ระบบ คู่มือออนไลน์

รายวิชา 060243110 JavaScript Programming

### จัดทำโดย

- |             |               |                              |
|-------------|---------------|------------------------------|
| 1.ธนภัทร    | แสงคำพันธุ์   | 6206021611028 (หัวหน้ากลุ่ม) |
| 2.จิรพัฒน์  | คำแท้         | 6206021621023                |
| 3.พีระวิชญ์ | พงศ์ประภากรณ์ | 6206021621155                |
| 4.ภัทรมน    | เมืองดี       | 6206021621121                |

นักศึกษาชั้นปีที่ 2 ตอนเรียนที่ 2

เสนอ

รองศาสตราจารย์ อนิราช มิ่งขวัญ

ภาควิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีและการจัดการอุตสาหกรรม ปีการศึกษา 2563 ภาคเรียนที่ 2

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ วิทยาเขตปทุมธานี

## คำนำ

รายงานเล่มนี้เป็นรายละเอียดการจัดทำระบบ คู่มือออนไลน์ ส่วนหนึ่งของวิชา 060243110 JavaScript Programming จัดทำขึ้นเพื่อแสดงศึกษาการเขียน โปรแกรมภาษาจาวา ในวิชาเรียน ซึ่ง เป็นการนำความรู้จากการเรียนภาคปฏิบัติในห้องเรียนมาประยุกต์ใช้ เพื่อให้เกิดความรู้ความเข้าใจในการเรียนมากขึ้น โดยใน รายงานเล่มนี้จะประกอบไปด้วย รายละเอียดของระบบงานการออกแบบระบบงาน การออกฐานข้อมูล เพื่อใช้งานร่วมกัน และออกแบบหน้าตาของโปรแกรมเพื่อมาประยุกต์ใช้

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า รายงานเล่มนี้จะก่อให้เกิดประโยชน์ แก่ผู้ที่สนใจศึกษาในเรื่องการเขียนโปรแกรมภาษาจาวาไม่มากนักน้อย แล้วถ้าหากมีข้อผิดพลาดประการใด ทางผู้จัดทำต้องขออภัยมา ณ ที่นี้ด้วย

คณะผู้จัดทำ

## สารบัญ

โปรแกรมระบบ ดูหนังสือออนไลน์.....	4
ซอฟต์แวร์ที่เกี่ยวข้องในการจัดทำโปรแกรม .....	5
Source Code .....	8
บรรณานุกรม .....	40

# โปรแกรมระบบดูหนังออนไลน์

ชื่อโปรแกรม

ระบบดูหนังออนไลน์

จัดทำโดย

- 1.ธนภัทร แสงคำพันธุ์ 6206021611028 (หัวหน้ากลุ่ม)
- 2.จิรพัฒน์ คำแท้ 6206021621023
- 3.พีระวิชญ์ พงศ์ประภากรณ์ 6206021621155
- 4.ภัทรมน เมืองดี 6206021621121

นักศึกษาชั้นปีที่ 2 ตอนเรียนที่ 2

## รายละเอียดโปรแกรม

เป็นโปรแกรมในการจัดการเกี่ยวกับการเข้าดูหนังในเว็บ สามารถเลือกดูหนังที่ต้องการได้และสามารถเพิ่มหรือแก้ไขได้

## ที่มาและความสำคัญของโปรแกรม

โปรแกรมระบบดูหนังออนไลน์ทางผู้จัดทำได้จัดทำขึ้นมาเพื่อ สร้างความรู้ความเข้าใจใน ภาษาจาวา ที่ได้จากการเรียนภาคทฤษฎีและภาคปฏิบัติเขียนโปรแกรมในห้องเรียน โดยมีการเขียนโปรแกรม ออกมาเป็นหน้าเว็บเพจ อีกทั้งยังสามารถเชื่อมต่อกับฐานข้อมูลต่างๆได้ โดยใช้โปรแกรม VS Code (Visual Studio Code)

## ซอฟต์แวร์ที่เกี่ยวข้องในการจัดทำโปรแกรม

### • MongoDB

MongoDB เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือ ไม่มี relation(ความสัมพันธ์) ของตารางแบบ SQL ทั่วไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทนการบันทึกข้อมูลทุกๆ record ใน MongoDB จะเรียกว่า Document ซึ่งจะเก็บค่าเป็น Key และ Value และการเก็บข้อมูล document ใน MongoDB จะ ถูกเก็บไว้ใน Collections (เปรียบเทียบกับ Table ใน Relation Database ทั่วไป)

### • POSTMAN

Postman คือ เครื่องมือสำหรับช่วยในการพัฒนา API ทดสอบการทำงานของ Service รวมถึงการ Mock Service อีกด้วย ซึ่งช่วยให้เราทำเรื่องยาก ให้กลายเป็นเรื่องง่าย

### • Nodejs

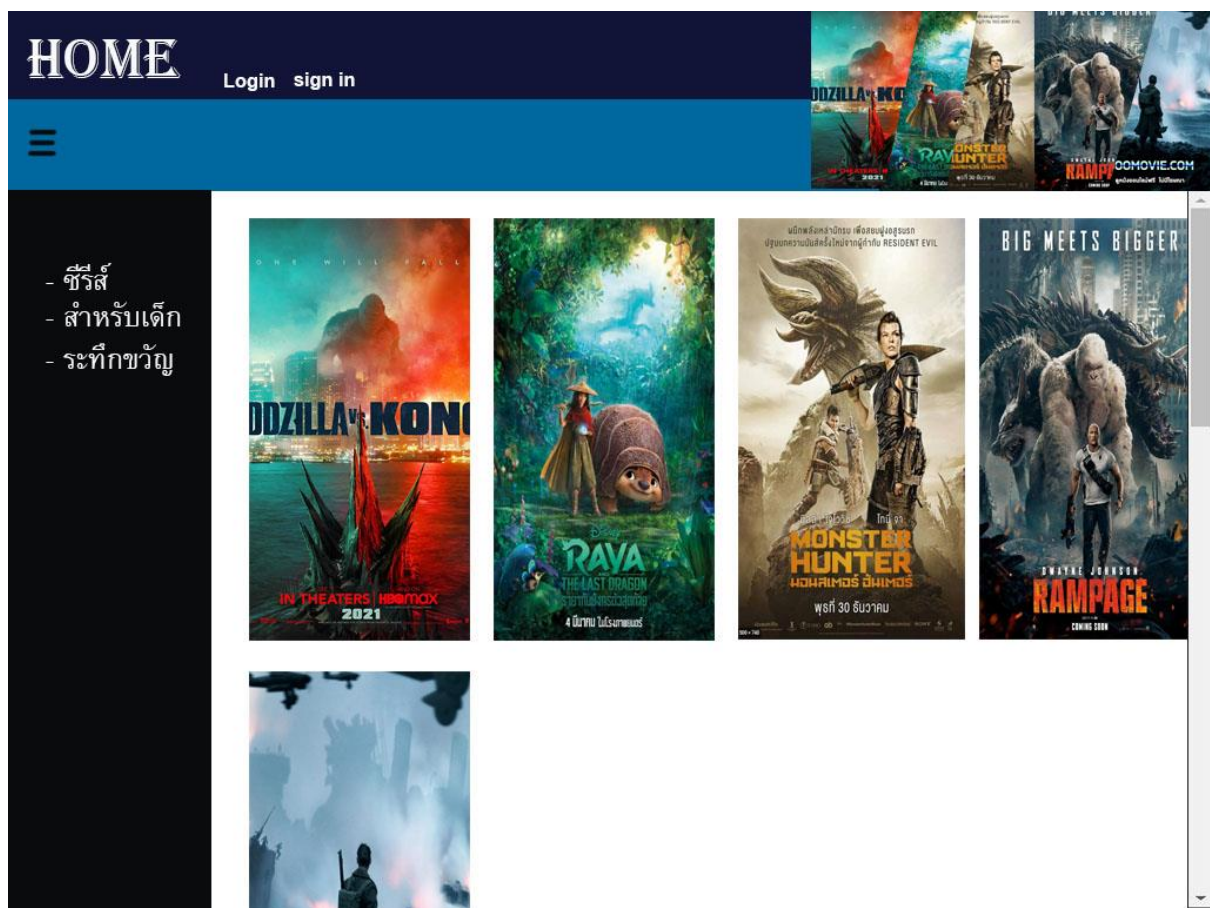
Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript สรุปคือ NodeJS คือ Platform ตัวหนึ่งที่ใช้เขียนด้วย JavaScript สำหรับเป็น Web Server

## ขั้นตอนการออกแบบระบบ

หน้าแรกเป็นหน้าเมนู มีรายละเอียดดังนี้

- มีปุ่มให้Log-in (เข้าสู่ระบบ)
- มีปุ่มให้Sign-in (สมัครสมาชิก)
- เข้าดูหนังตามต้องการ

หน้าแรก



Loin-Sign in

# Home

[Login](#)[Sign in](#)

## Login

username

password

login

sign in

## sign in

username

password

Confirm password

E-mail

Create an account

หน้า play

# HOME

[Login](#)[sign in](#)

- ซีรีส์

- สำหรับเด็ก

- ระทึกขวัญ

Monster Hunter

อนิเมะ

อนิเมะ

อนิเมะ

# Source Code

## FrontEnd

### 1.Index.html

```
index.html > html > head > link
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
8   <link rel="stylesheet" href="./style.css">
9   <title>ดูหนัง</title>
10 </head>
11 <body style="background-color: #1C2833;">
12   <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #396086; color: white;">
13     <div class="container-fluid">
14       <a class="navbar-brand" href="/Movie">Movie</a>
15       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false">
16         <span class="navbar-toggler-icon"></span>
17       </button>
18       <div class="collapse navbar-collapse" id="navbarSupportedContent">
19         <ul class="navbar-nav me-auto mb-2 mb-lg-0">
20           <li class="nav-item">
21             <a class="nav-link active" aria-current="page" href="/show/index.html">หน้าแรก</a>
22           </li>
23           <li class="nav-item dropdown">
24             <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
25               รายการหนังทั้งหมด
26             </a>
27             <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
28               <li><a class="dropdown-item" href="/play/play1.html">ดาบพิฆาตอสูร เดอะมูวี่ ศึกรถไฟสู่นิรันดร์</a></li>
29               <li><a class="dropdown-item" href="/play/play2.html">godzilla vs. kong</a></li>
30               <li><a class="dropdown-item" href="/play/play3.html">Gokushufudo</a></li>
31               <li><a class="dropdown-item" href="/play/play4.html">FAST & FURIOUS 9</a></li>
32               <li><a class="dropdown-item" href="/play/play5.html">อิเลventh</a></li>
33             </ul>
34           </li>
35           <li class="nav-item">
36             <a class="nav-link" href="/newmovie/new.html">เพิ่มหนัง</a>
37           </li>
38         </ul>
39       </div>
40     </div>
41   </nav>
42
43   <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
44     <div class="carousel-indicators">
45       <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0" class="active" aria-current="true" aria-label="Slide 1"></button>
46       <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1" aria-label="Slide 2"></button>
47       <button type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2" aria-label="Slide 3"></button>
48     </div>
49     <div class="carousel-inner">
50       <div class="carousel-item active">
51         
52       </div>
53       <div class="carousel-item">
54         
55       </div>
56       <div class="carousel-item">
57         
58       </div>
59     </div>
60     <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">
61       <span class="carousel-control-prev-icon" aria-hidden="true"></span>
62       <span class="visually-hidden">Previous</span>
63     </button>
64     <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
65       <span class="carousel-control-next-icon" aria-hidden="true"></span>
66       <span class="visually-hidden">Next</span>
67     </button>
68   </div>
69
70   <div class="container" id="anime" style="background-color: #1C2833; color: white; padding: 10px;">
71   </div>
72
73
```



```
73
74     <div class="card flex">
75         <div class="card-image" align = "center">
76             
77             <div class="cover" >
78                 <div class="img-overlay">
79                     <div class="studio">
80                     </div>
81                 </div>
82             </div>
83         </div>
84
85         <div class="card-content" align = "center" >
86
87             <h1 class='set-movie1' ></h1>
88
89             <div class="description">
90                 <br>
91                 <p class='set-movie11'></p>
92             </div>
93
94             <div class="cta">
95                 <a href="../play/play1.html">
96                     watch now!
97                 </a>
98             </div>
99
100         </div>
101     </div>
102
103
104
105     <br>
106
107
```

```
108 <div class="card flex " >
109   <div class="card-image" align = "center">
110     
111     <div class="cover">
112       <div class="img-overlay">
113         <div class="studio">
114         </div>
115       </div>
116     </div>
117   </div>
118
119   <div class="card-content" align = "center" >
120
121     <h1 class='set-movie2' ></h1>
122
123     <div class="description">
124       <br>
125       <p class='set-movie22'></p>
126     </div>
127
128     <div class="cta">
129       <a href="../play/play2.html">
130       watch now!
131     </a>
132     </div>
133
134   </div>
135
136 </div>
137
138
139
140 <br>
141
142
```

```
143 <div class="card flex">
144   <div class="card-image" align = "center">
145     
146     <div class="cover">
147       <div class="img-overlay">
148         <div class="studio">
149         </div>
150       </div>
151     </div>
152   </div>
153
154   <div class="card-content" align = "center" :
155
156     <h1 class='set-movie3'></h1>
157
158     <div class="description">
159       <br>
160       <p class='set-movie33'></p>
161     </div>
162
163     <div class="cta">
164       <a href="../play/play3.html">
165         watch now!
166       </a>
167     </div>
168
169   </div>
170
171 </div>
172
173
174 <br>
175
176
177
```

```
178 <div class="card flex">
179   <div class="card-image" align = "center">
180     
181     <div class="cover">
182       <div class="img-overlay">
183         <div class="studio">
184           </div>
185         </div>
186       </div>
187     </div>
188
189     <div class="card-content" align = "center" >
190
191       <h1 class='set-movie4'></h1>
192
193       <div class="description">
194         <br>
195         <p class='set-movie44'></p>
196       </div>
197
198       <div class="cta">
199         <a href="../play/play4.html">
200           watch now!
201         </a>
202       </div>
203
204     </div>
205   </div>
206 </div>
207
208
209 <br>
210
211
```

```
212 <div class="card flex">
213   <div class="card-image" align = "center">
214     
215     <div class="cover">
216       <div class="img-overlay">
217         <div class="studio">
218
219         </div>
220       </div>
221     </div>
222   </div>
223
224   <div class="card-content" align = "center" >
225
226     <h1 class='set-movie5'></h1>
227
228     <div class="description">
229       <br>
230       <p class='set-movie55'></p>
231     </div>
232
233     <div class="cta">
234       <a href="../../play/play5.html">
235         watch now!
236       </a>
237     </div>
238
239   </div>
240
241 </div>
242
243 <br>
```

```

246
247 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"
248 integrity="sha384-e03MYsd53ii+sc0/bJGfSiCZc+5NDVn2yr8+0RDqr0Ql0h+rpP48ckxlpbzkGwra6" crossorigin="anonymous">
249 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js"
250 integrity="sha384-JEW9xMcG8R+pH31jmWH6WP0WintQrMb4s7Z0dauHnUtxwog2vI5DkltS3qm9Ekf" crossorigin="anonymous"></script>
251
252 <script src='https://cdnjs.cloudflare.com/ajax/libs/vue/2.6.10/vue.min.js'></script>
253 <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js'></script>
254 <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
255 <script src="main.js"></script>
256 </body>
257 </html>

```

## play1.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-e03MYsd53ii+sc0/bJ
9 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WP0WintQ
10
11 </head>
12 <body>
13 <nav class="navbar navbar-expand-lg navbar-light bg-light" style="background-color: #396086!important;">
14 <div class="container-fluid">
15 <a class="navbar-brand" href=" ../index.html">Movie</a>
16 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSup
17 <span class="navbar-toggler-icon"></span>
18 </button>
19 <div class="collapse navbar-collapse" id="navbarSupportedContent">
20 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
21 <li class="nav-item">
22 <a class="nav-link active" aria-current="page" href=" ../index.html">หน้าแรก</a>
23 </li>
24
25 <li class="nav-item dropdown">
26 <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="fals
27 รายการหนังทั้งหมด
28 </a>
29 <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
30 <li><a class="dropdown-item" href=" ../play/play1.html">ดาบพิฆาตอสูร เดอะมูวี่ คิงออฟไ้สู้วันคร</a></li>
31 <li><a class="dropdown-item" href=" ../play/play2.html">godzilla vs. kong</a></li>
32 <li><a class="dropdown-item" href=" ../play/play3.html">Gokushufudo</a></li>
33 <li><a class="dropdown-item" href=" ../play/play4.html">FAST & FURIOUS 9</a></li>
34 <li><a class="dropdown-item" href=" ../play/play5.html">อีเชียนซิ่ง</a></li>
35 </ul>
36 </li>
37 </li>

```



new.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5
6      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8      <title>Document</title>
9    </head>
10
11    <body>
12      <form action="/newmovie/new.html" method="get">
13        <label for="uname"><b>namemovie</b></label>
14        <input id="name" type="text" placeholder="" name="name_" />
15
16        <label for="uname"><b>imgmovie</b></label>
17        <input type="text" placeholder="" id="img" name="img" />
18
19        <label for="psw"><b>vdo_ex</b></label>
20        <input type="text" placeholder="" id="ex" name="ex" />
21
22        <label for="uname"><b>vod_main</b></label>
23        <input type="text" placeholder="" name="mainvdo" id="mainvdo" />
24
25        <label for="image">upload image</label>
26        <input type="file" name="img" id="img">
27
28        <div class="form-group">
29          <a href="/" class="btn btn-warning">Cancel</a>
30          <button type="submit" class="btn btn-primary">Save</button>
31        </div>
32
33        <!--<button type="submit" class="clickkk">เพิ่ม</button>-->
34      </form>
35      <a href=javascript:history.back(1)>ย้อนกลับ</th></a>
36      <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
37      <script src="new.js"></script>
38
39    </body>
40  </html>
```



Main.js

```
1  let apiUser = "http://javap.app.ruk-com.cloud/api/users"
2  let apimovie = "http://javap.app.ruk-com.cloud/api/name-movie"
3
4  let setmovie1 = document.querySelector(".set-movie1");
5  let setmovie11 = document.querySelector(".set-movie11");
6
7  let setmovie2 = document.querySelector(".set-movie2");
8  let setmovie22 = document.querySelector(".set-movie22");
9
10 let setmovie3 = document.querySelector(".set-movie3");
11 let setmovie33 = document.querySelector(".set-movie33");
12
13 let setmovie4 = document.querySelector(".set-movie4");
14 let setmovie44 = document.querySelector(".set-movie44");
15
16 let setmovie5 = document.querySelector(".set-movie5");
17 let setmovie55 = document.querySelector(".set-movie55");
18
19 getmovie1()
20 getmovie11()
21
22 getmovie2()
23 getmovie22()
24
25 getmovie3()
26 getmovie33()
27
28 getmovie4()
29 getmovie44()
30
31 getmovie5()
32 getmovie55()
33
34 async function getmovie1(){
35     var movieObj = await axios.get(apimovie)
36     let picture = movieObj.data[0].name
37     setmovie1.innerHTML = picture
```

```
38 }
39 async function getmovie11(){
40     var movieObj = await axios.get(apimovie)
41     let story =movieObj.data[0].story
42     setmovie11.innerHTML = story
43
44 }
45
46
47 async function getmovie2(){
48
49     var movieObj = await axios.get(apimovie)
50     let picture =movieObj.data[1].name
51     setmovie2.innerHTML = picture
52
53 }
54 async function getmovie22(){
55     var movieObj = await axios.get(apimovie)
56     let story =movieObj.data[1].story
57     setmovie22.innerHTML = story
58
59 }
60
61
62
63 async function getmovie3(){
64
65     var movieObj = await axios.get(apimovie)
66     let picture =movieObj.data[2].name
67     setmovie3.innerHTML = picture
68
69 }
70 async function getmovie33(){
71     var movieObj = await axios.get(apimovie)
72     let story =movieObj.data[2].story
73     setmovie33.innerHTML = story
74
```

```
75     }
76
77     async function getmovie4(){
78
79         var movieObj = await axios.get(apimovie)
80         let picture =movieObj.data[3].name
81         setmovie4.innerHTML = picture
82
83     }
84     async function getmovie44(){
85         var movieObj = await axios.get(apimovie)
86         let story =movieObj.data[3].story
87         setmovie44.innerHTML = story
88
89     }
90
91
92     async function getmovie5(){
93
94         var movieObj = await axios.get(apimovie)
95         let picture =movieObj.data[4].name
96         setmovie5.innerHTML = picture
97
98     }
99     async function getmovie55(){
100         var movieObj = await axios.get(apimovie)
101         let story =movieObj.data[4].story
102         setmovie55.innerHTML = story
103
104     }
105
106
```

play/main.js

```
1  let apimovie = "http://javap.app.ruk-com.cloud/api/name-movie"
2  let setmovie1 = document.querySelector(".set-movie1");
3  let setmovie2 = document.querySelector(".set-movie2");
4  let setmovie3 = document.querySelector(".set-movie3");
5  let setmovie4 = document.querySelector(".set-movie4");
6  let setmovie5 = document.querySelector(".set-movie5");
7
8  getmovie1()
9  getmovie2()
10 getmovie3()
11 getmovie4()
12 getmovie5()
13
14 async function getmovie1(){
15     var movieObj = await axios.get(apimovie)
16     let picture =movieObj.data[0].name
17     setmovie1.innerHTML = picture
18 }
19
20 async function getmovie2(){
21
22     var movieObj = await axios.get(apimovie)
23     let picture =movieObj.data[1].name
24     setmovie2.innerHTML = picture
25 }
26
27 async function getmovie3(){
28
29     var movieObj = await axios.get(apimovie)
30     let picture =movieObj.data[2].name
31     setmovie3.innerHTML = picture
32 }
33
34 async function getmovie4(){
35     var movieObj = await axios.get(apimovie)
36     let picture =movieObj.data[3].name
37     setmovie4.innerHTML = picture
```

```

38
39 }
40 async function getmovie5(){
41
42     var movieObj = await axios.get(apimovie)
43     let picture =movieObj.data[4].name
44     setmovie5.innerHTML = picture
45
46 }

```

new.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8" />
5
6          <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8          <title>Document</title>
9      </head>
10
11      <body>
12          <form action="/newmovie/new.html" method="get">
13              <label for="uname"><b>namemovie</b></label>
14              <input id="name" type="text" placeholder="" name="name_" />
15
16              <label for="uname"><b>imgmovie</b></label>
17              <input type="text" placeholder="" id="img" name="img" />
18
19              <label for="psw"><b>vdo_ex</b></label>
20              <input type="text" placeholder="" id="ex" name="ex" />
21
22              <label for="uname"><b>vod_main</b></label>
23              <input type="text" placeholder="" name="mainvdo" id="mainvdo" />
24
25              <label for="image">upload image</label>
26              <input type="file" name="img" id="img">
27
28              <div class="form-group">
29                  <a href="/" class="btn btn-warning">Cancel</a>
30                  <button type="submit" class = "btn btn-primary">Save</button>
31              </div>
32
33              <!--<button type="submit" class="clickkk">เพิ่ม</button>-->
34          </form>
35          <a href=javascript:history.back(1)>ย้อนกลับ</th></a>
36          <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
37          <script src="new.js"></script>

```

```
37     <script src="new.js"></script>
38   </body>
39 </html>
```

showname.js

```
1  let apimovie = "http://javap.app.ruk-com.cloud/api/name-movie"
2  let setmovie1 = document.querySelector(".movie1");
3  let setmovie2 = document.querySelector(".movie2");
4  let setmovie3 = document.querySelector(".movie3");
5  let setmovie4 = document.querySelector(".movie4");
6  let setmovie5 = document.querySelector(".movie5");
7
8  getmovie1()
9  getmovie2()
10 getmovie3()
11 getmovie4()
12 getmovie5()
13
14 async function getmovie1(){
15     var movieObj = await axios.get(apimovie)
16     let picture = movieObj.data[0].name
17     setmovie1.innerHTML = picture
18 }
19
20 async function getmovie2(){
21
22     var movieObj = await axios.get(apimovie)
23     let picture = movieObj.data[1].name
24     setmovie2.innerHTML = picture
25 }
26
27 async function getmovie3(){
28
29     var movieObj = await axios.get(apimovie)
30     let picture = movieObj.data[2].name
31     setmovie3.innerHTML = picture
32 }
33
34 async function getmovie4(){
35     var movieObj = await axios.get(apimovie)
36     let picture = movieObj.data[3].name
37     setmovie4.innerHTML = picture
```

```

38
39 }
40 async function getmovie5(){
41
42     var movieObj = await axios.get(apimovie)
43     let picture =movieObj.data[4].name
44     setmovie5.innerHTML = picture
45
46 }

```

## Backend

### Index.js

```

1  var express = require("express");
2  var cors = require("cors");
3  var bodyParser = require("body-parser");
4  var mongoose = require("mongoose");
5
6  var mongo_uri = 'mongodb+srv://admin:8QnwGw3c8yFnAFLU@cluster0.jixbg.mongodb.net/myFirstDatabase?retryWrites=true&w=majority';
7  mongoose.Promise = global.Promise;
8  mongoose.connect(mongo_uri, { useNewUrlParser: true ,useUnifiedTopology: true}).then(
9      () => {
10         console.log("[success] task 2 : connected to the database ");
11     },
12     (error) => {
13         console.log("[failed] task 2 " + error);
14         process.exit();
15     }
16 );
17
18 var app = express();
19
20 app.use(cors());
21 app.use(bodyParser.urlencoded({ extended: true }));
22 app.use(bodyParser.json());
23
24 var port = process.env.PORT || 5000;
25
26 app.listen(port, () => {
27     console.log("[success] task 1 : listening on port " + port);
28 });
29
30 app.get("/", (req, res) => {
31     res.status(200).send("หน้าแรกของ api express");
32 });
33
34 app.post('/', (req, res)=>{
35
36     console.log(JSON.stringify(req.fields));
37 });

```

```

38
39 // path สำหรับ MongoDB ของเรา
40 var users = require("./router/users_router");
41 var movie = require("./router/movie_router");
42
43 app.use("/api/users", users);
44 app.use("/api/name-movie", movie);
45
46 app.use((req, res, next) => {
47     var err = new Error("ไม่พบ path ที่คุณต้องการ");
48     err.status = 404;
49     next(err);
50 });
51
52
53

```

.htaccess

```

⚙ .htaccess
1 <IfModule mod_rewrite.c>
2     RewriteEngine On
3     RewriteBase /
4     RewriteRule ^index.html$ - [L]
5     RewriteCond %{REQUEST_FILENAME} !-f
6     RewriteCond %{REQUEST_FILENAME} !-d
7     RewriteRule . /index.html [L]
8 </IfModule>

```



movie\_router.js

```
router > JS movie_router.js > ...
1  var express = require("express");
2  const bodyParser = require('body-parser')
3  var router = express.Router();
4  const multer = require('multer')
5  var movie = require("../model/movie_model");
6
7  const { request } = require("express");
8
9
10 const storage = multer.diskStorage({
11   destination: function (request,file,callback){
12     |   callback(null, './newmovie/upload/images');
13   },
14
15   filename: function(request,file,callback){
16     |   callback(null,Date.now()+file.originalname);
17   },
18 });
19
20
21 const upload = multer({
22   storage: storage,
23   limits: {
24     |   fileSize: 1024 * 1024 * 3,
25   },
26 });
27
28
29 // GET all
30 router.get("/", (req, res) => {
31   |   movie.find().exec((err, data) => {
32     |   if (err) return res.status(400).send(err);
33     |   res.status(200).send(data);
34   |   });
35 });
36
```

```

37 // GET 1
38 router.get("/:id", (req, res) => {
39     movie.findById(req.params._id).exec((err, data) => {
40         if (err) return res.status(400).send(err);
41         res.status(200).send(data);
42     });
43 });
44
45 // POST (create new data)
46 router.post("/", upload.single('image') , async (req, res) => {
47     console.log(request.file);
48     const objForInsert = new movie({
49         name: req.body.name,
50         img: req.body.img,
51         vdo_ex: req.body.vdo_ex,
52         vdo_main: req.body.vdo_main,
53         img:request.file.filename,
54     });
55
56     objForInsert.save((err, data) => {
57         if (err) return res.status(400).send(err);
58         res.status(200).send("เพิ่มข้อมูลเรียบร้อยแล้ว");
59     });
60 });
61
62 // PUT (update current data)
63 router.put("/:id", (req, res) => {
64     movie.findByIdAndUpdate(req.params._id, req.body, (err, data) => {
65         if (err) return res.status(400).send(err);
66         res.status(200).send("อัปเดตข้อมูลเรียบร้อยแล้ว");
67     });
68 });
69
70 // DELETE (delete 1 data)
71 router.delete("/:id", (req, res) => {
72     movie.findByIdAndDelete(req.params._id, (err, data) => {
73         if (err) return res.status(400).send(err);

```

```

74         res.status(200).send("ลบข้อมูลเรียบร้อยแล้ว");
75     });
76 });
77
78 module.exports = router;
79

```

users\_router.js

```
router > JS users_router.js > router.post("/") callback
1  var express = require("express");
2  var router = express.Router();
3  var user = require("../model/users_model");
4
5  // GET all
6  router.get("/", (req, res) => {
7    user.find().exec((err, data) => {
8      if (err) return res.status(400).send(err);
9      res.status(200).send(data);
10   });
11 });
12
13 // GET 1
14 router.get("/:_id", (req, res) => {
15   user.findById(req.params._id).exec((err, data) => {
16     if (err) return res.status(400).send(err);
17     res.status(200).send(data);
18   });
19 });
20
21 // POST (create new data)
22 router.post("/", (req, res) => {
23   var obj = new user(req.body);
24   obj.save((err, data) => {
25     if (err) return res.status(400).send(err);
26     res.status(200).send("เพิ่มข้อมูลเรียบร้อยแล้ว");
27   });
28 });
29
30 // PUT (update current data)
31 router.put("/:_id", (req, res) => {
32   user.findByIdAndUpdate(req.params._id, req.body, (err, data) => {
33     if (err) return res.status(400).send(err);
34     res.status(200).send("อัปเดตข้อมูลเรียบร้อยแล้ว");
35   });
36 });
37
```

```

38 // DELETE (delete 1 data)
39 router.delete("/:id", (req, res) => {
40     user.findByIdAndDelete(req.params._id, (err, data) => {
41         if (err) return res.status(400).send(err);
42         res.status(200).send("ลบข้อมูลเรียบร้อยแล้ว");
43     });
44 });
45
46 module.exports = router;
47

```

new.js

```

newmovie > JS new.js > clickNext.addEventListener("click") callback > app.post("/") callback
1 let apimovie = "http://javap.app.ruk-com.cloud/api/name-movie/";
2 let clickNext = document.querySelector(".clickkk");
3 const { default: axios } = require("axios");
4 var movie = require("../model/movie_model");
5 var mongoose = require("mongoose");
6
7 mongoose.connect('mongodb+srv://admin:8QnwGw3c8yFnAFLU@cluster0.jixbg.mongodb.net/myFirstDatabase?retryWrites=true&w=majority')
8 var Schema = new mongoose.Schema({
9     name: String,
10    img: String,
11    vdo_ex: String,
12    vdo_main: String,
13 })
14
15 var user = mongoose.model('name-movie', Schema)
16 // post();
17
18 clickNext.addEventListener("click", function () {
19     let namemovie = document.getElementById("name").value;
20     let img = document.getElementById("img").value;
21     let ex = document.getElementById("ex").value;
22     let mainvdo = document.getElementById("mainvdo").value;
23
24     app.post('/', function(req, res){
25         new movie({
26             name: req.namemovie,
27             img: req.img,
28             vdo_ex: req.ex,
29             vdo_main: req.mainvdo,
30         }).save(function(err, doc){
31             if(err) res.json(err);
32             else res.send('Successfully inserted')
33         })
34     });
35
36

```

```
37 // axios.post("http://movie.app.ruk-com.cloud/api/name-movie",{
38 //   name: namemovie,
39 //   img: img,
40 //   vdo_ex: ex,
41 //   vdo_main: mainvdo,
42 // })
43
44
45
46
47 alert(namemovie + "<<<<");
48 // axios({
49 //   method: "post",
50 //   url: apimovie,
51 //   data: {
52 //     name: name,
53 //     img: img,
54 //     vdo_ex: ex,
55 //     vdomain: mainvdo,
56 //   },
57 // });
58
59 // let name = document.getElementById("name").value;
60 // let img = document.getElementById("img").value;
61 // let ex = document.getElementById("ex").value;
62 // let mainvdo = document.getElementById("mainvdo").value;
63 // let res = await axios.post(apimovie, {
64 //   name: name_,
65 //   img: img,
66 //   vdo_ex: ex,
67 //   vdo_main: mainvdo,
68 // });
69 // let data = res.data;
70 // alert(data);
71 });
72
```

```

72
73 // axios.post(
74 //   "http://movie.app.ruk-com.cloud/api/name-movie",
75 //   {
76 //     name: document.getElementById("name"),
77 //     img: document.getElementById("img"),
78 //     vdo_ex: document.getElementById("ex"),
79 //     vdo_main: document.getElementById("mainvdo"),
80 //   },
81 //   {
82 //     // Config
83 //   }
84 // );
85
86 // axios
87 //   .post(apimovie, obj)
88 //   .then(function (response) {
89 //     console.log("yeser");
90 //   })
91 //   .catch(function (error) {
92 //     console.log(error);
93 //   });<a href=javascript:history.back(1)>ย้อนกลับ</th></a>

```

movie\_model.js

```

model > JS movie_model.js > ...
1  var mongoose = require("mongoose");
2
3  var movie_schema = mongoose.Schema(
4    {
5      name: {
6        type: String,
7      },
8      img: {
9        type: String,
10     },
11     vdo_ex: {
12       type: String,
13     },
14     vdo_main: {
15       type: String,
16     },
17   }, {
18     collection: "name-movie",
19   });
20  var name_movie = mongoose.model("name-movie", movie_schema)
21
22  module.exports = name_movie;
23

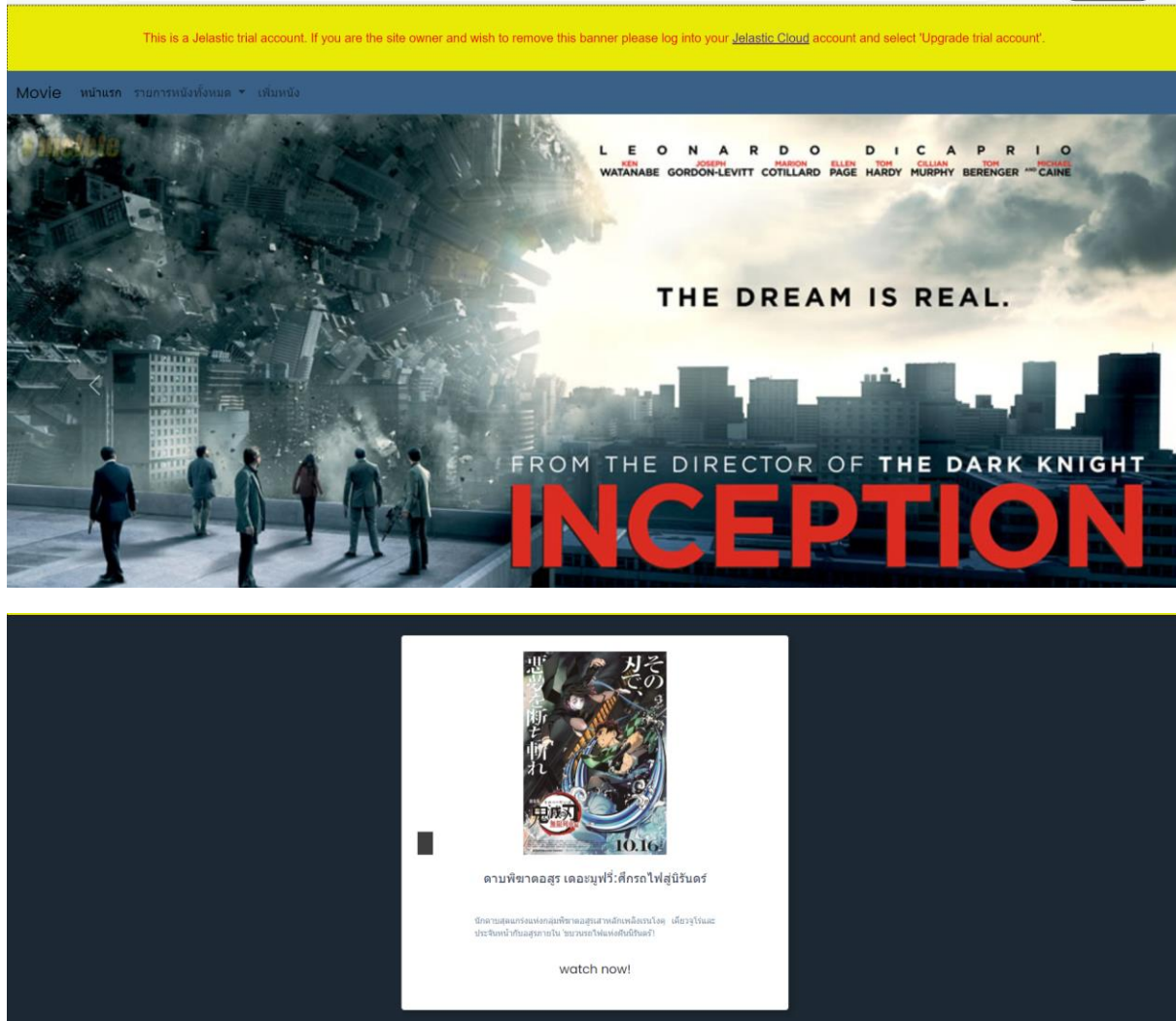
```

users\_model.js

```
model > JS users_model.js > ...
1  var mongoose = require("mongoose");
2
3  var user_schema = mongoose.Schema(
4    {
5      email: {
6        type: String,
7      },
8      user: {
9        type: String,
10     },
11     password: {
12       type: String,
13     },
14   },
15   {
16     collection: "users",
17   });
18
19  var users = mongoose.model("users", user_schema);
20
21  module.exports = users;
22
23
```

## แสดงตัวอย่างหน้าเว็บ

หน้าHOME







### Godzilla vs. Kong

ศึกครั้งใหญ่ที่ระหว่างสองยักษ์ใหญ่ "Godzilla vs. Kong - ก๊อดซิลล่า ปะทะ คอง" มหาภาพยนตร์มหัศจรรย์ใหม่ในจักรวาลภาพยนตร์มอนสเตอร์เวิร์ลส์ของเลเจนดารี โดยกำกับของ อัดัม วีนการ์ด

watch now!



### Gokushufudo

เป็นเรื่องราวของ Tatsu ชายผู้เป็นยากูซ่าระดับตำนานที่คนในวงการต่างเกรงกลัว แต่อยู่มาวันหนึ่งเขาได้หายตัวไปอย่างลึกลับและไม่มีใครรู้ว่าเขาหายไปไหน จนถูกเล่าลือกันไปต่างๆนาๆบ้างก็ว่าเขาถูกอุ้มหรือแอบหลบหนีตำรวจ

watch now!



## FAST & FURIOUS 9

แอ็คชั่นจะขยายไปทั่วโลก จากลอนดอนสู่โตเกียว จากอเมริกากลางสู่  
เอตินเบอร์ระ จากหลุมหลบภัยลับในอาเซอร์ไบจาน สู่ถนนในทบิลีซี  
เพื่อนเก่ากลับมามีชีวิตอีกครั้ง

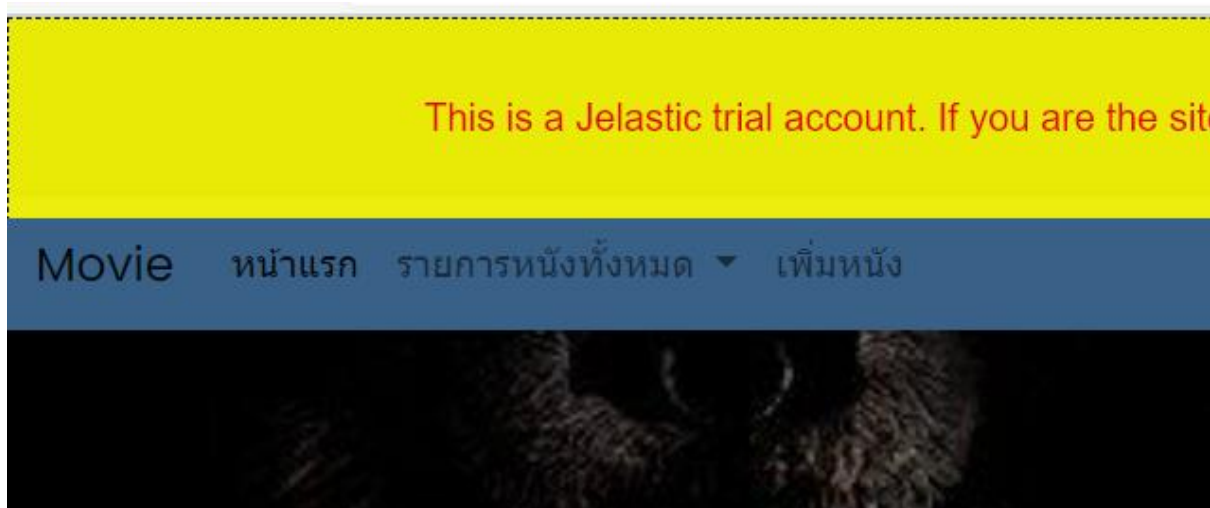
watch now!



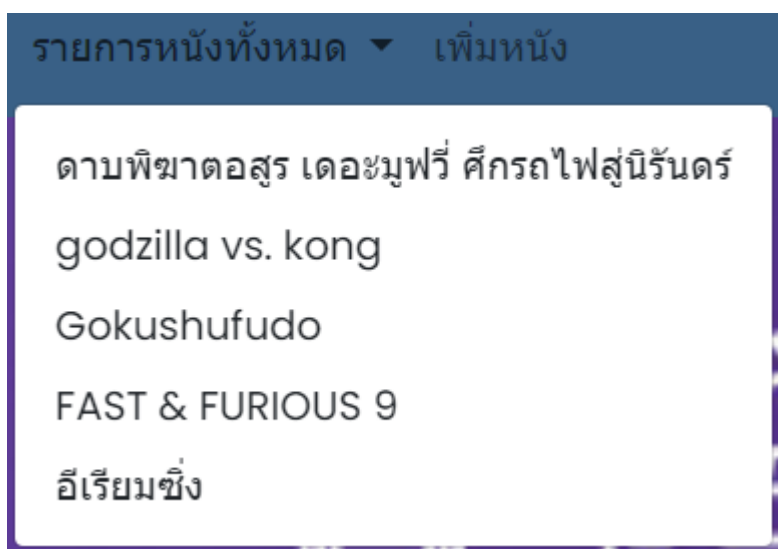
## อีเรียมซิ่ง

เบลล่า นำทัพความฮา มาพร้อมน้ำคอม ขอให้ฟ้าเป็นสีฟ้า ขอให้น้ำเป็นสีเขียว ขอให้เฮฮาร่าเริงกันพร้อมเพรียง ขอให้ฝันอีเรียมเป็นจริง ภาพยนตร์ Adventure-Comedy ว่าด้วยเรื่องของ “เรียม” (เบลล่า ราณี) เจ้าแม่ตัวอิจฉาประจำหมู่บ้านบางน้ำกร่อย

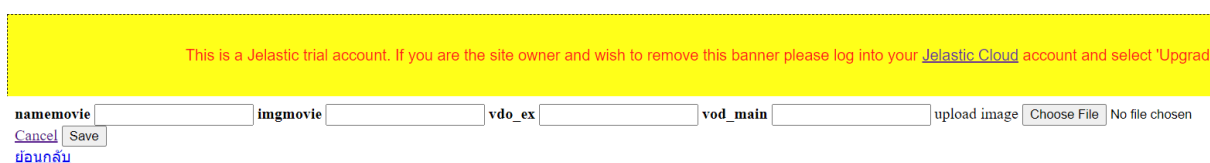
watch now!



กด Movie หรือ หน้าแรก จะเป็นการกดกลับมาได้เดิม



รายการหนังทั้งหมดจะโชว์หนังทั้งหมดที่มีในเว็บไซต์และสามารถกดผ่านตามรายชื่อหนังเพื่อรับชมได้เลย



หน้าเพิ่มหนังจะมีให้ใส่ชื่อหนัง ชื่อรูป วิดีโอตัวอย่าง วิดีโอหลัก และเลือกรูปภาพ และกด save หรือ cancel

ตรงย้อนกลับกดไปแล้วจะกลับไปหน้า Movie

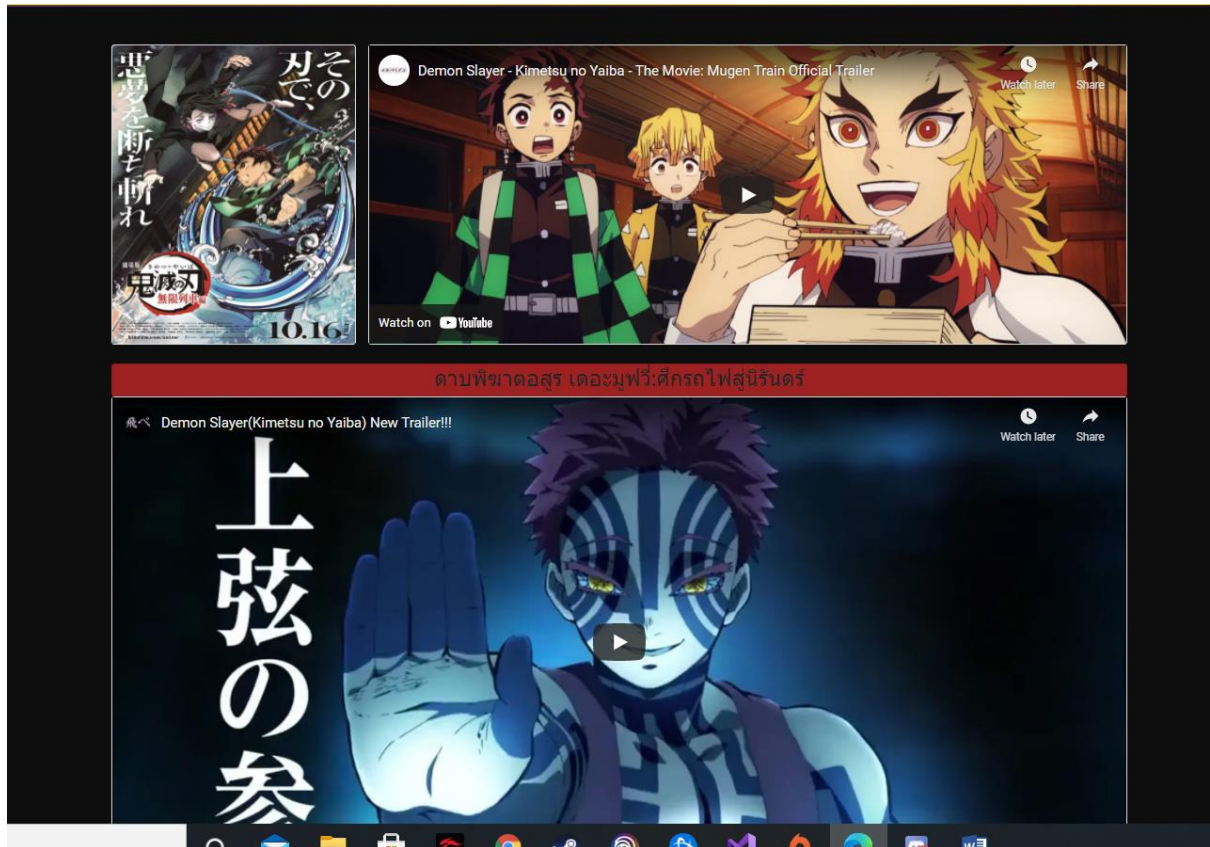


## ดาบพิฆาตอสูร เดอะมูฟวี่: ศีกรถไฟสู่นิรันดร์

นักดาบสุดแกร่งแห่งกลุ่มพิฆาตอสูรเส้าหลักเพลิงเรนโจคุ เคียวจูโร่และ  
ประจันหน้ากับอสูรภายใน 'ขบวนรถไฟแห่งนิรันดร์'!

watch now!

แสดง รูปภาพและบทย่อของหนัง ถ้ากด watch now!จะเป็นการเข้าหน้า play



หน้า play จะมีรูปหนัง วิดีโอตัวอย่างกับวิดีโอหลัก สามารถเล่นวิดีโอได้ บนหัววิดีโอจะมีชื่อเรื่องแสดง

## บรรณานุกรม

- 1) MongoDB คืออะไร ? ผู้เขียน Chai Phonbopit วันที่ 16 สิงหาคม 2558  
<https://devahoy.com/blog/2015/08/getting-started-with-mongodb/>
- 2) Vue.js คืออะไร ? สอนการใช้งาน Vue.js ฉบับเริ่มต้น ผู้เขียน Chai Phonbopit วันที่ 4 สิงหาคม 2562  
<https://devahoy.com/blog/2019/08/introduction-to-vuejs/>
- 3) มาทดสอบ API ด้วย PostMan กันเถอะ ผู้เขียน Ae Eiei วันที่ 2 กันยายน 2560  
<https://medium.com/@nest12345/%E0%B8%A1%E0%B8%B2%E0%B8%97%E0%B8%94%E0%B8%AA%E0%B8%AD%E0%B8%9A-api-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-postman-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-c750a09c86ce>
- 4) NodeJS คืออะไร? เริ่มต้นใช้งาน NodeJS ผู้เขียน Chai Phonbopit วันที่ 31 กรกฎาคม 2558 <https://devahoy.com/blog/2015/07/getting-started-with-nodejs/>
- 5) npm run serv <https://youtu.be/vr6O-IYebXa>

