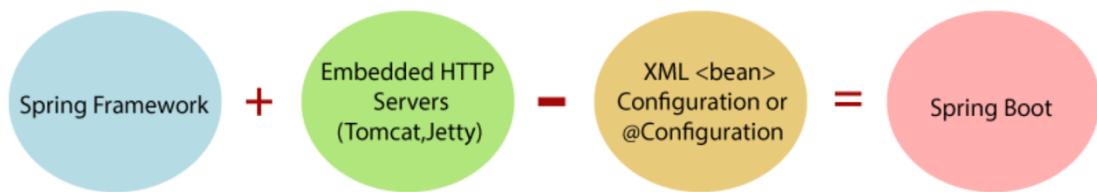


Java Spring Boot

Spring Boot Framework គឺជាប្រព័ន្ធឌែលបង្កើតដោយ “The Spring Team” ដើម្បីបង្របន្ទូយការ Development នៃ Spring Application ដូច កាត់ប្រើនឹង Spring Boot Application ត្រូវការ Spring configuration តិចត្លូចណាស់។



យោត្តិឱ្យរាល់បាយពីរបៀបនេះ Spring Boot?

ប្រព័ន្ធឌែលបង្កើតនិងអាមេរិកការ Development នៃ Web Application ។

សារម្មមេរាល់នៅ Spring Boot គឺសុវត្ថិភាព គុណភាព ហើយ Project ដែលបង្កើតដោយ Spring Boot គឺមានភាពរហូតដើរលើនឹងនិងមាន MVC មកដាមួយត្រូវបានដំបូងត្រូវតែម៉ឺន់ៗ

JDK version 8

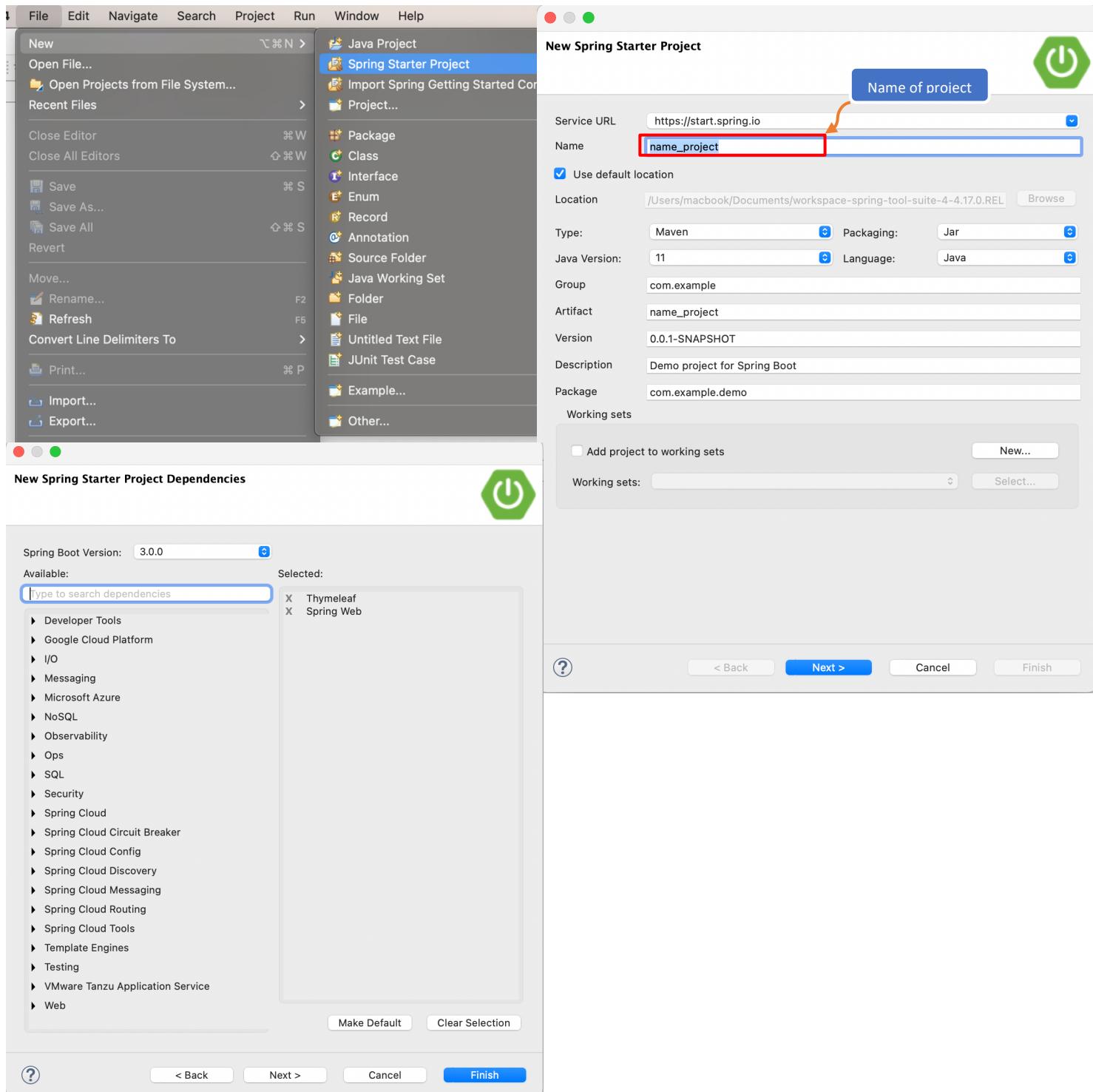
<https://www.oracle.com/java/technologies/downloads/>

IDE មានដូចជា

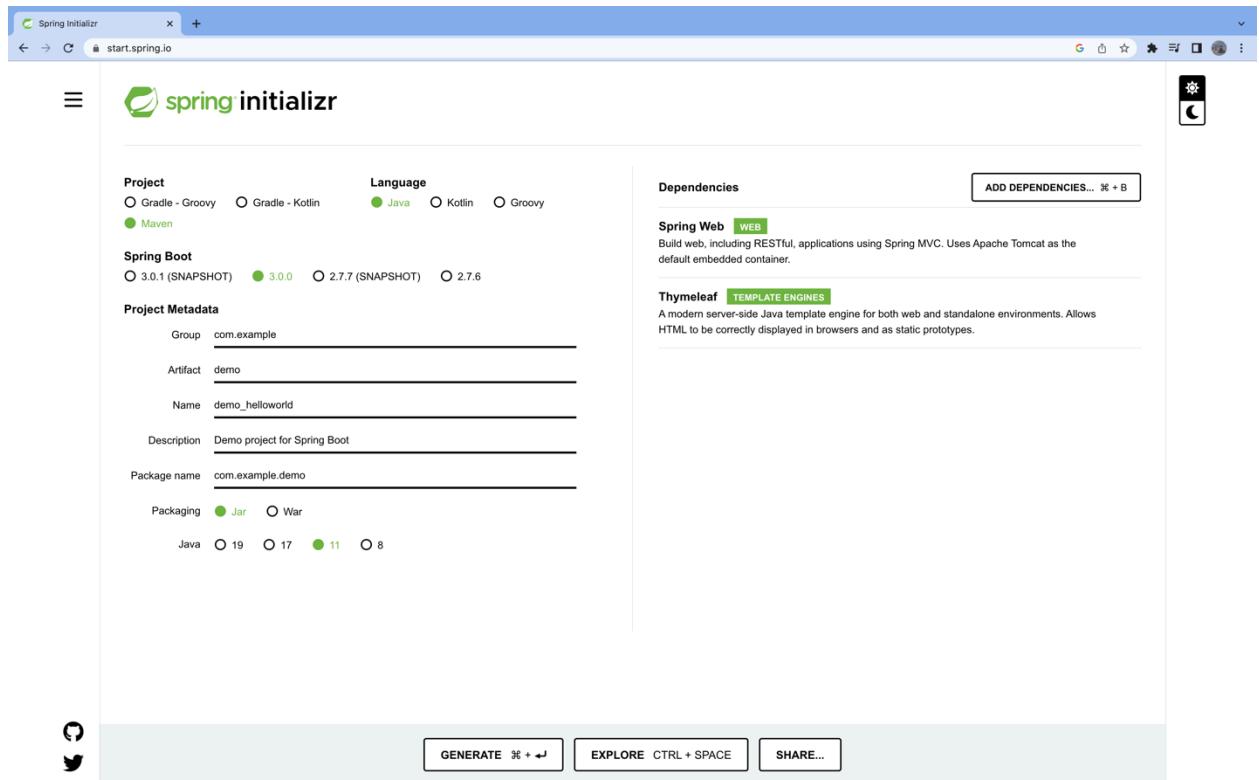


កម្រិតបង្កើត Spring boot Project

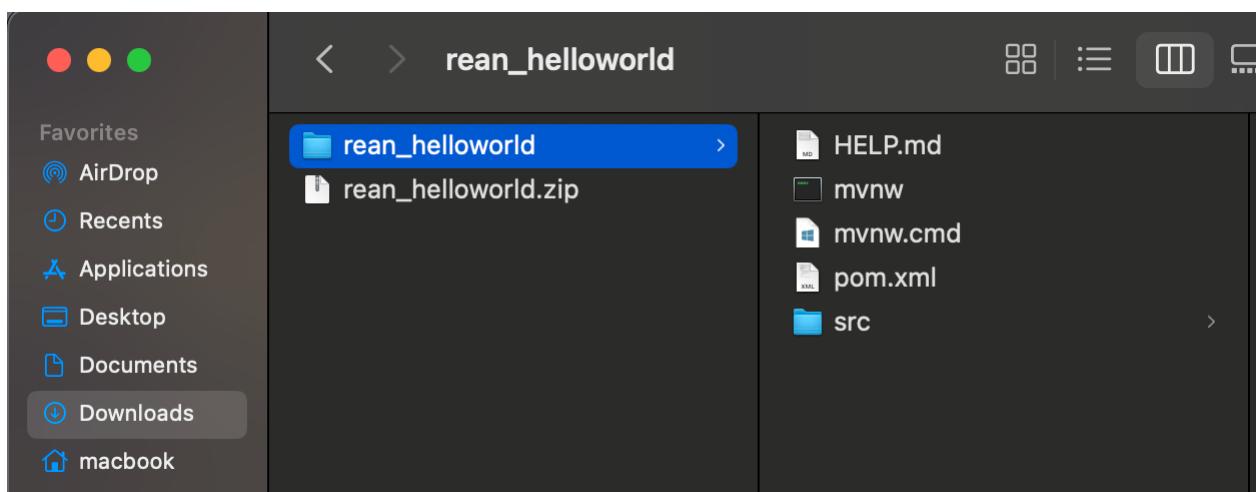
របៀបទឹនយបដើតនៅក្នុង Spring ToolSuite



របៀបទិន្នន័យការកើតឡើង Website (Link : <https://start.spring.io>)

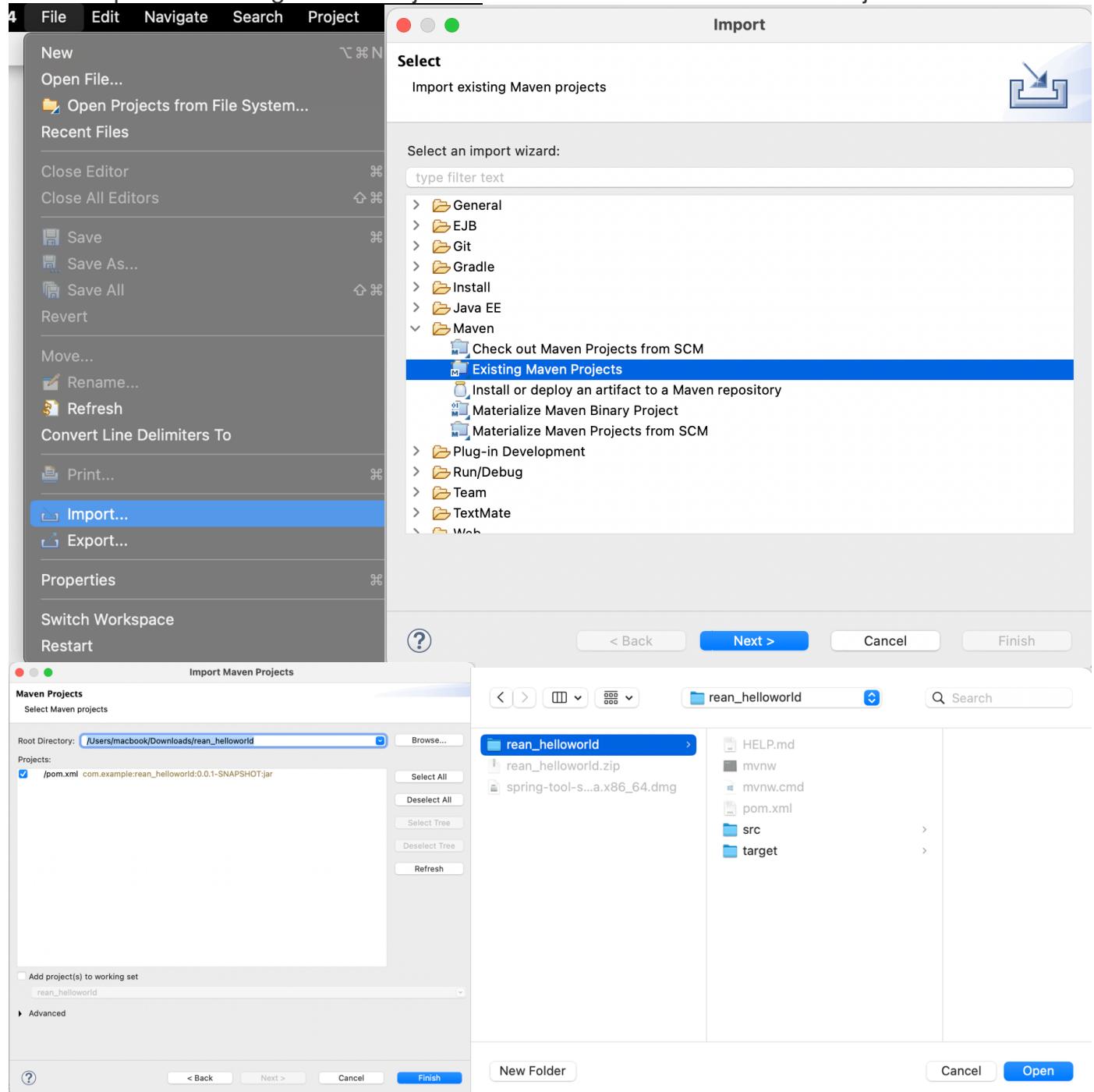


បន្ទាប់ពីបូប Generate គឺទទួលបាន File Zip មួយនិង Folder Project មួយ



Import Project in Spring ToolSuite

File -> Import -> Existing Maven Project -> Next -> Browse -> Select the Project Folder

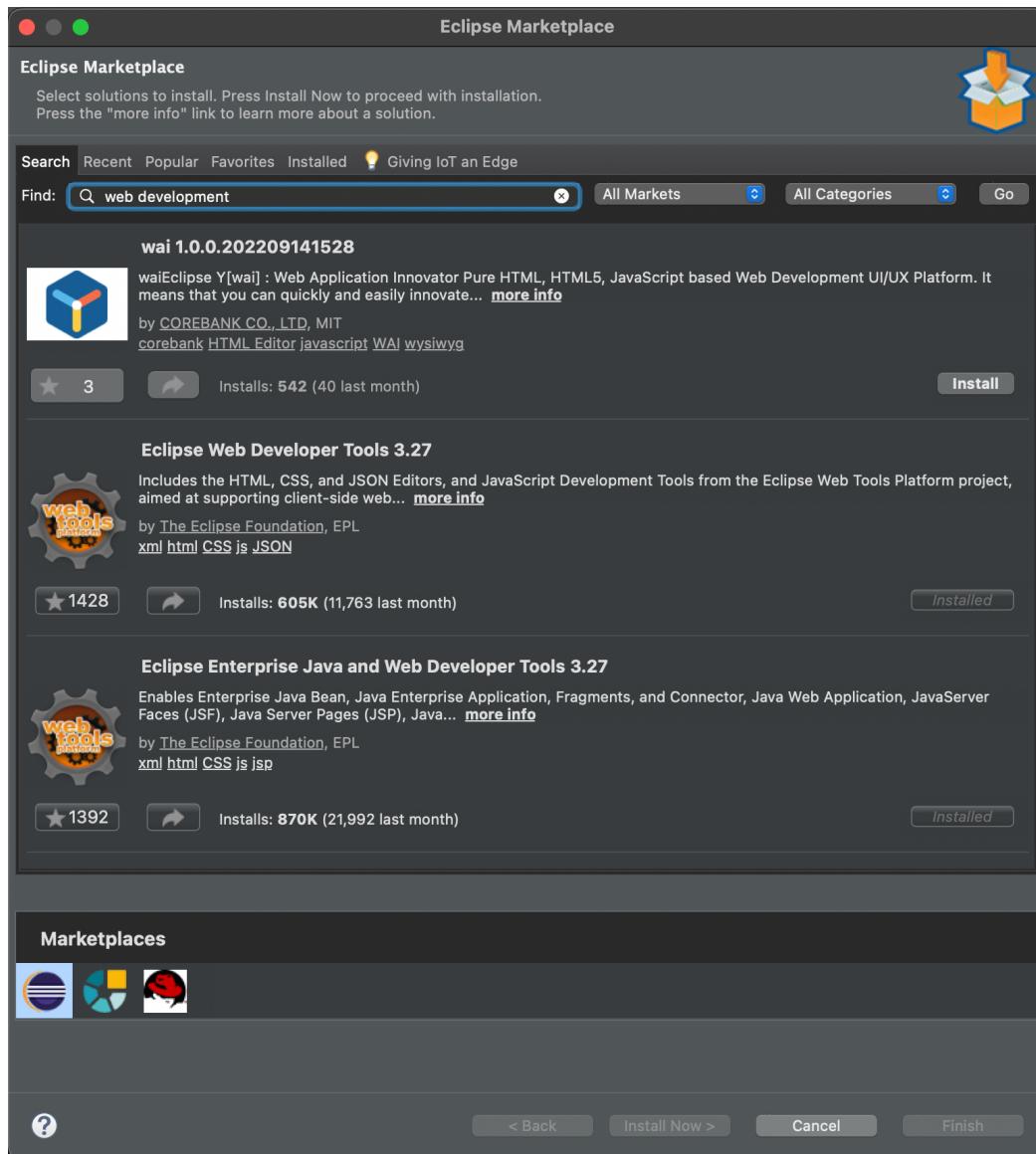


Spring Boot กំរើត Hello world

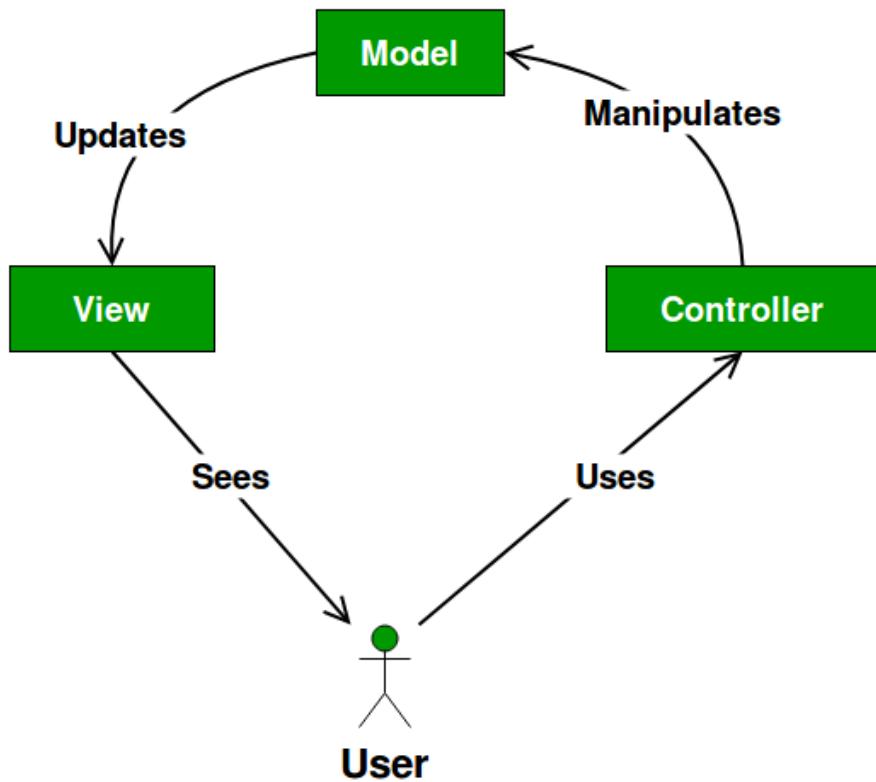
ដើម្បីសរសេរការណា Web ផ្តល់ជាមួយ HTML, CSS, Bootstrap, JavaScript...

បានយើងត្រូវចូលទៅក្នុង Help ->Eclipse Marketplace បន្ទាប់មកទេរទៀត

Search ពាក្យ Web Development រូបភាព Install



MVC = Model View Controller



សំគាល់

គ្នាំនៃ MVC

docs.spring.io/spring-boot/docs/current/reference/html/index.html

មាននៃ MVC

javatpoint.com/spring-boot-thymeleaf-view

Thymeleaf Dependency

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

ឧចាបរណ៍ ១

The screenshot shows the Spring Tool Suite (STS) interface. The top left contains the code for `DemoHelloworldApplication.java`, which defines a Spring Boot application with a single endpoint returning "index". The bottom left contains the content of `index.html`. The top right shows the Spring Boot application's logs in the terminal, indicating successful startup and deployment. The bottom right shows a browser window displaying the application's home page with the title "Hello World" and the message "Welcome to Spring Boot".

```
1 package com.example.demo;
2
3 import org.springframework.boot.SpringApplication;
4
5 @SpringBootApplication
6 @Controller
7 public class DemoHelloworldApplication {
8
9     public static void main(String[] args) {
10         SpringApplication.run(DemoHelloworldApplication.class, args);
11     }
12
13     @GetMapping("/")
14     public String home() {
15         return "index";
16     }
17 }
18
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
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
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
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
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
60410
60411
60412
60413
60414
60415
60416
60417
60418
60419
60420
60421
60422
60423
60424
60425
60426
60427
60428
60429
60430
60431
60432
60433
60434
60435
60436
60437
60438
60439
60440
60441
60442
60443
60444
60445
60446
60447
60448
60449
60450
60451
60452
60453
60454
60455
60456
60457
60458
60459
60460
60461
60462
60463
60464
60465
60466
60467
60468
60469
60470
60471
60472
60473
60474
60475
60476
60477
60478
60479
60480
60481
60482
60483
60484
60485
60486
60487
60488
60489
604810
604811
604812
604813
604814
604815
604816
604817
604818
604819
604820
604821
604822
604823
604824
604825
604826
604827
604828
604829
604830
604831
604832
604833
604834
604835
604836
604837
604838
604839
6048310
6048311
6048312
6048313
6048314
6048315
6048316
6048317
6048318
6048319
6048320
6048321
6048322
6048323
6048324
6048325
6048326
6048327
6048328
6048329
6048330
6048331
6048332
6048333
6048334
6048335
6048336
6048337
6048338
6048339
60483310
60483311
60483312
60483313
60483314
60483315
60483316
60483317
60483318
60483319
60483320
60483321
60483322
60483323
60483324
60483325
60483326
60483327
60483328
60483329
60483330
60483331
60483332
60483333
60483334
60483335
60483336
60483337
60483338
60483339
604833310
604833311
604833312
604833313
604833314
604833315
604833316
604833317
604833318
604833319
604833320
604833321
604833322
604833323
604833324
604833325
604833326
604833327
604833328
604833329
604833330
604833331
604833332
604833333
604833334
604833335
604833336
604833337
604833338
604833339
6048333310
6048333311
6048333312
6048333313
6048333314
6048333315
6048333316
6048333317
6048333318
6048333319
6048333320
6048333321
6048333322
6048333323
6048333324
6048333325
6048333326
6048333327
6048333328
6048333329
6048333330
6048333331
6048333332
6048333333
6048333334
6048333335
6048333336
6048333337
6048333338
6048333339
60483333310
60483333311
60483333312
60483333313
60483333314
60483333315
60483333316
60483333317
60483333318
60483333319
60483333320
60483333321
60483333322
60483333323
60483333324
60483333325
60483333326
60483333327
60483333328
60483333329
60483333330
60483333331
60483333332
60483333333
60483333334
60483333335
60483333336
60483333337
60483333338
60483333339
604833333310
604833333311
604833333312
604833333313
604833333314
604833333315
604833333316
604833333317
604833333318
604833333319
604833333320
604833333321
604833333322
604833333323
604833333324
604833333325
604833333326
604833333327
604833333328
604833333329
604833333330
604833333331
604833333332
604833333333
604833333334
604833333335
604833333336
604833333337
604833333338
604833333339
6048333333310
6048333333311
6048333333312
6048333333313
6048333333314
6048333333315
6048333333316
6048333333317
6048333333318
6048333333319
6048333333320
6048333333321
6048333333322
6048333333323
6048333333324
6048333333325
6048333333326
6048333333327
6048333333328
6048333333329
6048333333330
6048333333331
6048333333332
6048333333333
6048333333334
6048333333335
6048333333336
6048333333337
6048333333338
6048333333339
60483333333310
60483333333311
60483333333312
60483333333313
60483333333314
60483333333315
60483333333316
60483333333317
60483333333318
60483333333319
60483333333320
60483333333321
60483333333322
60483333333323
60483333333324
60483333333325
60483333333326
60483333333327
60483333333328
60483333333329
60483333333330
60483333333331
60483333333332
60483333333333
60483333333334
60483333333335
60483333333336
60483333333337
60483333333338
60483333333339
604833333333310
604833333333311
604833333333312
604833333333313
604833333333314
604833333333315
604833333333316
604833333333317
604833333333318
604833333333319
604833333333320
604833333333321
604833333333322
604833333333323
604833333333324
604833333333325
604833333333326
604833333333327
604833333333328
604833333333329
604833333333330
604833333333331
604833333333332
604833333333333
604833333333334
604833333333335
604833333333336
604833333333337
604833333333338
604833333333339
6048333333333310
6048333333333311
6048333333333312
6048333333333313
6048333333333314
6048333333333315
6048333333333316
6048333333333317
6048333333333318
6048333333333319
6048333333333320
6048333333333321
6048333333333322
6048333333333323
6048333333333324
6048333333333325
6048333333333326
6048333333333327
6048333333333328
6048333333333329
6048333333333330
6048333333333331
6048333333333332
6048333333333333
6048333333333334
6048333333333335
6048333333333336
6048333333333337
6048333333333338
6048333333333339
60483333333333310
60483333333333311
60483333333333312
60483333333333313
60483333333333314
60483333333333315
60483333333333316
60483333333333317
60483333333333318
60483333333333319
60483333333333320
60483333333333321
60483333333333322
60483333333333323
60483333333333324
60483333333333325
60483333333333326
60483333333333327
60483333333333328
60483333333333329
60483333333333330
60483333333333331
60483333333333332
60483333333333333
60483333333333334
60483333333333335
60483333333333336
60483333333333337
60483333333333338
60483333333333339
604833333333333310
604833333333333311
604833333333333312
604833333333333313
604833333333333314
604833333333333315
604833333333333316
604833333333333317
604833333333333318
604833333333333319
604833333333333320
604833333333333321
604833333333333322
604833333333333323
604833333333333324
604833333333333325
604833333333333326
604833333333333327
604833333333333328
604833333333333329
604833333333333330
604833333333333331
604833333333333332
604833333333333333
604833333333333334
604833333333333335
604833333333333336
604833333333333337
604833333333333338
604833333333333339
6048333333333333310
6048333333333333311
6048333333333333312
6048333333333333313
6048333333333333314
6048333333333333315
6048333333333333316
6048333333333333317
6048333333333333318
6048333333333333319
6048333333333333320
6048333333333333321
6048333333333333322
6048333333333333323
6048333333333333324
6048333333333333325
6048333333333333326
6048333333333333327
6048333333333333328
6048333333333333329
6048333333333333330
6048333333333333331
6048333333333333332
6048333333333333333
6048333333333333334
6048333333333333335
6048333333333333336
6048333333333333337
6048333333333333338
6048333333333333339
60483333333333333310
60483333333333333311
60483333333333333312
60483333333333333313
60483333333333333314
60483333333333333315
60483333333333333316
60483333333333333317
60483333333333333318
60483333333333333319
60483333333333333320
60483333333333333321
60483333333333333322
60483333333333333323
60483333333333333324
60483333333333333325
60483333333333333326
60483333333333333327
60483333333333333328
60483333333333333329
60483333333333333330
60483333333333333331
60483333333333333332
60483333333333333333
60483333333333333334
60483333333333333335
60483333333333333336
60483333333333333337
60483333333333333338
60483333333333333339
604833333333333333310
604833333333333333311
604833333333333333312
604833333333333333313
604833333333333333314
604833333333333333315
604833333333333333316
604833333333333333317
604833333333333333318
604833333333333333319
604833333333333333320
604833333333333333321
604833333333333333322
604833333333333333323
604833333333333333324
604833333333333333325
604833333333333333326
604833333333333333327
604833333333333333328
604833333333333333329
604833333333333333330
604833333333333333331
604833333333333333332
604833333333333333333
604833333333333333334
604833333333333333335
604833333333333333336
604833333333333333337
604833333333333333338
604833333333333333339
6048333333333333333310
6048333333333333333311
6048333333333333333312
6048333333333333333313
6048333333333333333314
6048333333333333333315
6048333333333333333316
6048333333333333333317
6048333333333333333318
6048333333333333333319
6048333333333333333320
6048333333333333333321
6048333333333333333322
6048333333333333333323
6048333333333333333324
6048333333333333333325
6048333333333333333326
6048333333333333333327
6048333333333333333328
6048333333333333333329
6048333333333333333330
6048333333333333333331
6048333333333333333332
6048333333333333333333
6048333333333333333334
6048333333333333333335
6048333333333333333336
6048333333333333333337
6048333333333333333338
6048333333333333333339
60483333333333333333310
60483333333333333333311
60483333333333333333312
60483333333333333333313
60483333333333333333314
60483333333333333333315
60483333333333333333316
60483333333333333333317
60483333333333333333318
60483333333333333333319
60483333333333333333320
60483333333333333333321
60483333333333333333322
60483333333333333333323
60483333333333333333324
60483333333333333333325
60483333333333333333326
60483333333333333333327
60483333333333333333328
60483333333333333333329
60483333333333333333330
60483333333333333333331
60483333333333333333332
60483333333333333333333
60483333333333333333334
60483333333333333333335
60483333333333333333336
60483333333333333333337
60483333333333333333338
60483333333333333333339
604833333333333333333310
604833333333333333333311
604833333333333333333312
604833333333333333333313
604833333333333333333314
604833333333333333333315
604833333333333333333316
604833333333333333333317
604833333333333333333318
604833333333333333333319
604833333333333333333320
604833333333333333333321
604833333333333333333322
604833333333333333333323
604833333333333333333324
604833333333333333333325
604833333333333333333326
604833333333333333333327
604833333333333333333328
604833333333333333333329
604833333333333333333330
604833333333333333333331
604833333333333333333332
604833333333333333333333
604833333333333333333334
604833333333333333333335
604833333333333333333336
604833333333333333333337
604833333333333333333338
604833333333333333333339
6048333333333333333333310
6048333333
```

ເຫັນ Controller ກໍ່ບານໄດ້ຮັບໃຈຈະ Spring Boot ມານ MVC ປູ້ອໍານວຍ

The screenshot shows a Java Spring Boot application setup. On the left, the `Controller.java` file is displayed:

```
1 package com.example.demo.controller;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4
5 public class Controller {
6
7     @GetMapping
8     public String views() {
9         return "index";
10    }
11 }
12
```

On the right, the `index.html` file is shown:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Introduction</title>
6 </head>
7 <body>
8     <h3>Hello World <br>
9     In String boot have MVC ready</h3>
10
11 </body>
12 </html>
```

Below these files is a browser window titled "Introduction" showing the output of the application at `localhost:8080`:

Hello World
In String boot have MVC ready

ການເບີວິທີ່ແຈ້ງຍົດ Backend to Frontend

The screenshot shows a Java code editor and a browser window. In the code editor, `MyController.java` contains:

```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller
4 @Controller
5 public class MyController {
6
7     @GetMapping("/value")
8     public String views(Model model) {
9         int a=10,b=20;
10        model.addAttribute("a", a);
11        model.addAttribute("b", b);
12        return "index";
13    }
14
15 }
16 }
```

In the browser, the URL `localhost:8080/value` is shown, and the page displays:

Value A = 10
Value B = 20

The `index.html` file in the browser's developer tools shows the generated HTML:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Introduction</title>
6 </head>
7 <body>
8     <h3 th:text="' Value A = ' + ${a}"></h3>
9     <h3 th:text="' Value B = ' + ${b}"></h3>
10 </body>
11 </html>
```

ການໃສ່ກໍລະກົດປົວເຕັມໃນ Frontend

The screenshot shows a Java code editor and a browser window. In the code editor, `MyController.java` contains:

```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller
4 @Controller
5 public class MyController {
6
7     @GetMapping("/value")
8     public String views(Model model) {
9         int a=100,b=20;
10        model.addAttribute("a", a);
11        model.addAttribute("b", b);
12        return "index";
13    }
14
15 }
16 }
```

In the browser, the URL `localhost:8080/value` is shown, and the page displays:

Value A = 100
Value B = 20
Max is :100

The `index.html` file in the browser's developer tools shows the generated HTML with Thymeleaf expressions:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Introduction</title>
6 </head>
7 <body>
8     <h3 th:text="' Value A = ' + ${a}"></h3>
9     <h3 th:text="' Value B = ' + ${b}"></h3>
10    <div th:if="${a>b}">
11        <h3 th:text="'Max is :'+ ${a}"></h3>
12    </div>
13    <div th:unless="${a>b}">
14        <h3 th:text="'Min is :'+ ${b}"></h3>
15    </div>
16 </body>
17 </html>
```

Example:

Dependencies in project

```
20<dependencies>
21    <dependency>
22        <groupId>org.springframework.boot</groupId>
23        <artifactId>spring-boot-starter-thymeleaf</artifactId>
24    </dependency>
25    <dependency>
26        <groupId>org.springframework.boot</groupId>
27        <artifactId>spring-boot-starter-web</artifactId>
28    </dependency>
29
30    <dependency>
31        <groupId>org.projectlombok</groupId>
32        <artifactId>lombok</artifactId>
33        <optional>true</optional>
34    </dependency>
35    <dependency>
36        <groupId>org.springframework.boot</groupId>
37        <artifactId>spring-boot-starter-tomcat</artifactId>
38        <scope>provided</scope>
39    </dependency>
40    <dependency>
41        <groupId>org.springframework.boot</groupId>
42        <artifactId>spring-boot-starter-test</artifactId>
43        <scope>test</scope>
44    </dependency>
45</dependencies>
```

Class Model

```
1 package com.example.demo.model;
2
3 public class MyModel {
4
5     private String firstname;
6     private String lastname;
7     private String email;
8     private String address;
9
10    public MyModel() {
11        // TODO Auto-generated constructor stub
12    }
13    public MyModel(String firstname, String lastname, String email, String address) {
14        super();
15        this.firstname = firstname;
16        this.lastname = lastname;
17        this.email = email;
18        this.address = address;
19    }
20    public String getFirstname() {
21        return firstname;
22    }
23    public void setFirstname(String firstname) {
24        this.firstname = firstname;
25    }
26    public String getLastname() {
27        return lastname;
28    }
29    public void setLastname(String lastname) {
30        this.lastname = lastname;
31    }
32    public String getEmail() {
33        return email;
34    }
35    public void setEmail(String email) {
36        this.email = email;
37    }
38    public String getAddress() {
39        return address;
40    }
41    public void setAddress(String address) {
42        this.address = address;
43    }
44 }
```

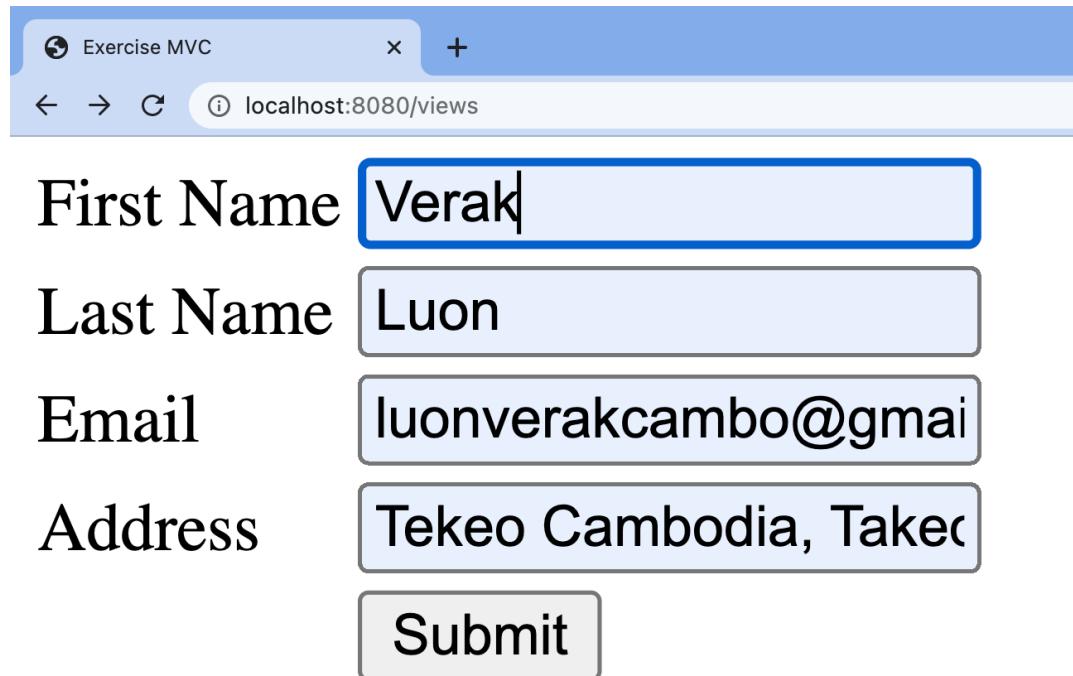
Class Controller

```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.GetMapping;
5 import org.springframework.web.bind.annotation.ModelAttribute;
6 import org.springframework.web.bind.annotation.RequestMapping;
7 import org.springframework.web.bind.annotation.RequestMethod;
8 import org.springframework.web.servlet.ModelAndView;
9
10 import com.example.demo.model.MyModel;
11
12 @Controller
13 public class MyController {
14
15     @GetMapping("/views")
16     public String home() {
17         return "index";
18     }
19
20     @RequestMapping(value = "/save", method = RequestMethod.POST)
21     public ModelAndView save(@ModelAttribute MyModel model)
22     {
23         ModelAndView modelAndView = new ModelAndView();
24         modelAndView.setViewName("user-data");
25         modelAndView.addObject("user", model);
26         return modelAndView;
27     }
28 }
```

Viesw

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Exercise MVC</title>
6 </head>
7 <body>
8 <form action="save" method="post"> | 
9   <table>
10    <tr>
11      <td><label for="firstname">First Name</label></td>
12      <td><input type="text" name="firstname"></input></td>
13    </tr>
14    <tr>
15      <td><label for="lastname">Last Name</label></td>
16      <td><input type="text" name="lastname"></input></td>
17    </tr>
18    <tr>
19      <td><label for="email">Email</label></td>
20      <td><input type="text" name="email"></input></td>
21    </tr>
22    <tr>
23      <td><label for="address">Address</label></td>
24      <td><input type="text" name="address"></input></td>
25    </tr>
26    <tr>
27      <td></td>
28      <td><input type="submit" value="Submit"></input></td>
29    </tr>
30  </table>
31 </form>
32 </body>
33 </html>
```

Result



The screenshot shows a web browser window with the title "Exercise MVC". The address bar indicates the URL is "localhost:8080/views". The page displays a form with four input fields and a submit button. The first field is labeled "First Name" and contains the value "Verak". The second field is labeled "Last Name" and contains the value "Luon". The third field is labeled "Email" and contains the value "luonverakcambo@gmail.com". The fourth field is labeled "Address" and contains the value "Tekeo Cambodia, Takeo". A blue rectangular box highlights the "First Name" field.

First Name	Verak
Last Name	Luon
Email	luonverakcambo@gmail.com
Address	Tekeo Cambodia, Takeo
Submit	

```
1<html xmlns:th="https://thymeleaf.org">
2  <table>
3    <tr>
4      <td><h4>Full Name: </h4></td>
5      <td><h4 th:text="${user.firstname} +' '+ ${user.lastname}"></h4></td>
6    </tr>
7    <tr>
8      <td><h4>Email ID: </h4></td>
9      <td><h4 th:text="${user.email}"></h4></td>
10   </tr>
11   <tr>
12     <td><h4>Address: </h4></td>
13     <td><h4 th:text="${user.address}"></h4></td>
14   </tr>
15 </table>
16 </html>
```

Result



Full Name: Verak Luon

E-mail : luonverakcambo@gmail.com

Address : Tekeo Cambodia

Crud Operation with MVC

Dependencies

```
20⑩    <dependencies>
21⑪        <dependency>
22            <groupId>org.springframework.boot</groupId>
23            <artifactId>spring-boot-starter-data-jpa</artifactId>
24        </dependency>
25        <dependency>
26            <groupId>org.springframework.boot</groupId>
27            <artifactId>spring-boot-starter-thymeleaf</artifactId>
28        </dependency>
29        <dependency>
30            <groupId>org.springframework.boot</groupId>
31            <artifactId>spring-boot-starter-web</artifactId>
32        </dependency>
33
34        <dependency>
35            <groupId>org.springframework.boot</groupId>
36            <artifactId>spring-boot-devtools</artifactId>
37            <scope>runtime</scope>
38            <optional>true</optional>
39        </dependency>
40        <dependency>
41            <groupId>com.mysql</groupId>
42            <artifactId>mysql-connector-j</artifactId>
43            <scope>runtime</scope>
44        </dependency>
45        <dependency>
46            <groupId>org.springframework.boot</groupId>
47            <artifactId>spring-boot-starter-tomcat</artifactId>
48            <scope>provided</scope>
49        </dependency>
50        <dependency>
51            <groupId>org.springframework.boot</groupId>
52            <artifactId>spring-boot-starter-test</artifactId>
53            <scope>test</scope>
54        </dependency>
55        <dependency>
56            <groupId>commons-io</groupId>
57            <artifactId>commons-io</artifactId>
58            <version>2.8.0</version>
59        </dependency>
60        <dependency>
61            <groupId>org.projectlombok</groupId>
62            <artifactId>lombok</artifactId>
63
64            <scope>provided</scope>
65        </dependency>
66    </dependencies>
```

Backend

```
1 package com.example.demo;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.boot.CommandLineRunner;
5 import org.springframework.boot.SpringApplication;
6 import org.springframework.boot.autoconfigure.SpringBootApplication;
7 import com.example.demo.repository.StudentRepository;
8
9 @SpringBootApplication
10 public class StudentApplication implements CommandLineRunner {
11
12     public static void main(String[] args) {
13
14         SpringApplication.run(StudentApplication.class, args);
15     }
16
17     @Autowired
18     private StudentRepository studentRepository;
19
20     @Override
21     public void run(String... args) throws Exception {
22
23 //         Student stu1=new Student("Luon", "Verak", "luonverak@gmail.com");
24 //         studentRepository.save(stu1);
25 //
26 //         Student stu2=new Student("Sok", "Cheata", "sokcheata@gmail.com");
27 //         studentRepository.save(stu2);
28 //
29 //         Student stu3=new Student("Seng", "Lyna", "senglyna@gmail.com");
30 //         studentRepository.save(stu3);
31 //
32     }
33
34 }
```

Class Controller

```
2@import org.springframework.stereotype.Controller;
10 @Controller
11 public class StudentController {
12
13     private StudentService studentService;
14
15     public StudentController(StudentService studentService) {
16         super();
17         this.studentService = studentService;
18     }
19     @GetMapping("/students")
20     public String listStudents(Model mod) {
21         mod.addAttribute("students",studentService.getAllStudents());
22         return "students";
23     }
24     @GetMapping("/students/new")
25     public String createStudentForm(Model mod) {
26
27         Student student=new Student();
28         mod.addAttribute("student", student);
29         return "create_student";
30     }
31     @PostMapping("/students")
32     public String saveStudent(@ModelAttribute ("student") Student student) {
33         studentService.saveStudent(student);
34         return "redirect:/students";
35     }
36     @GetMapping("/students/edit/{id}")
37     public String editStudentForm(@PathVariable Long id,Model mod) {
38         mod.addAttribute("student", studentService.getStudentById(id));
39         return "edit_student";
40     }
41     @PostMapping("/students/{id}")
42     public String updateStudent(@PathVariable Long id,@ModelAttribute("student") Student student,Model mod) {
43         Student existingStudent = studentService.getStudentById(id);
44         existingStudent.setId(id);
45         existingStudent.setFirstname(student.getFirstname());
46         existingStudent.setLastname(student.getLastname());
47         existingStudent.setEmail(student.getEmail());
48         existingStudent.setScore(student.getScore());
49         studentService.updateStudent(existingStudent);
50         return "redirect:/students";
51     }
52     @GetMapping("/students/{id}")
53     public String deleteStudent(@PathVariable Long id) {
54         studentService.deleteStudentById(id);
55         return "redirect:/students";
56     }
57 }
```

```
1 package com.example.demo.entity;
2+import jakarta.persistence.Column;[]
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @Getter
15 @Setter
16 @Entity
17 @Table(name="students")
18 public class Student {
19@   @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     private long id;
22@   @Column(name = "first_name", nullable = false)
23     private String firstname;
24@   @Column(name = "last_name", nullable = false)
25     private String lastname;
26@   @Column(name = "email", nullable = false)
27     private String email;
28@   @Column(name="score",nullable = false)
29     private float score;
1 package com.example.demo.repository;
2
3+import org.springframework.data.jpa.repository.JpaRepository;
4
5 import com.example.demo.entity.Student;
6
7 public interface StudentRepository extends JpaRepository<Student, Long>{
8
9 }
```

```
StudentService.java X
1 package com.example.demo.service;
2
3+import java.util.List;
4
5 import com.example.demo.entity.Student;
6
7 public interface StudentService {
8     List<Student> getAllStudents();
9     Student saveStudent(Student student);
10    Student getStudentById(Long id);
11    Student updateStudent(Student student);
12    void deleteStudentById(Long id);
13 }
```

```
1 package com.example.demo.service.impl;
2
3 import java.util.List;
4
5 import org.springframework.stereotype.Service;
6
7 import com.example.demo.entity.Student;
8 import com.example.demo.repository.StudentRepository;
9 import com.example.demo.service.StudentService;
10 @Service
11 public class StudentImp implements StudentService{
12
13     private StudentRepository studentRepository;
14
15     public StudentImp(StudentRepository studentRepository) {
16         super();
17         this.studentRepository = studentRepository;
18     }
19
20     @Override
21     public List<Student> getAllStudents() {
22
23         return studentRepository.findAll();
24     }
25
26     @Override
27     public Student saveStudent(Student student) {
28
29         return studentRepository.save(student);
30     }
31
32     @Override
33     public Student getStudentById(Long id) {
34         // TODO Auto-generated method stub
35         return studentRepository.findById(id).get();
36     }
37
38     @Override
39     public Student updateStudent(Student student) {
40         // TODO Auto-generated method stub
41         return studentRepository.save(student);
42     }
43
44     @Override
45     public void deleteStudentById(Long id) {
46         studentRepository.deleteById(id);
47
48     }
49
50
51 }
```

Student GUI

<https://getbootstrap.com/docs/4.0/getting-started/introduction/>

https://www.w3schools.com/bootstrap/bootstrap_navbar.asp

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Student Management</title>
6   <link rel="stylesheet"
7     href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
8     integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
9     crossorigin="anonymous">
10 </head>
11 <body>
12
13   <nav class="navbar navbar-expand-md bg-dark navbar-dark">
14     <!-- Brand -->
15     <a class="navbar-brand" th:href="@{/students}">Students Management</a>
16
17     <!-- Toggler/collapsible Button -->
18     <button class="navbar-toggler" type="button" data-toggle="collapse"
19       data-target="#collapsibleNavbar">
20       <span class="navbar-toggler-icon"></span>
21     </button>
22
23     <!-- Navbar links -->
24     <div class="collapse navbar-collapse" id="collapsibleNavbar">
25       <ul class="navbar-nav">
26         <li class="nav-item"><a class="nav-link" th:href="@{/students}">Students
27           Management</a></li>
28       </ul>
29     </div>
30   </nav>
31
32   <div class="container">
33     <div class="row">
34       <h1>List of Students</h1>
35     </div>
36
37     <div class="row">
38       <div class="col-lg-3">
39         <a th:href="@{/students/new}" class="btn btn-primary btn-sm mb-3">
40           Add Student</a>
41       </div>
42     </div>
43
44     <table class="table table-striped table-bordered">
45       <thead class="table-dark">
46         <tr>
47           ...
48       </tr>
49     </thead>
50     <tbody>
51       ...
52     </tbody>
53   </div>
```

Content-Type:html

Writable

```
26          <li class="nav-item"><a class="nav-link" th:href="@{/students}">Students  
27              Management</a></li>  
28      </ul>  
29  </div>  
30 </nav>  
31  
32  <div class="container">  
33      <div class="row">  
34          <h1>List of Students</h1>  
35      </div>  
36  
37      <div class="row">  
38          <div class="col-lg-3">  
39              <a th:href="@{/students/new}" class="btn btn-primary btn-sm mb-3">  
40                  Add Student</a>  
41          </div>  
42      </div>  
43  
44  <table class="table table-striped table-bordered">  
45      <thead class="table-dark">  
46          <tr>  
47              <th>First Name</th>  
48              <th>Last Name</th>  
49              <th>Email</th>  
50              <th>Score</th>  
51              <th>Actions</th>  
52          </tr>  
53      </thead>  
54      <tbody>  
55          <tr th:each="student: ${students}">  
56              <td th:text="${student.firstname}"></td>  
57              <td th:text="${student.lastname}"></td>  
58              <td th:text="${student.email}"></td>  
59              <td th:text="${student.score}"></td>  
60  
61              <td><a th:href="@{/students/edit/{id}(id=${student.id})}"  
62                  class="btn btn-primary">Update</a> <a  
63                  th:href="@{/students/{id}(id=${student.id})}"  
64                  class="btn btn-danger">Delete</a></td>  
65          </tr>  
66      </tbody>  
67  </table>  
68 </div>  
69  
70 </body>  
71 </html>
```

Create New Student

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Student Management</title>
6   <link rel="stylesheet"
7     href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
8     integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
9     crossorigin="anonymous">
10 </head>
11 <body>
12
13   <nav class="navbar navbar-expand-md bg-dark navbar-dark">
14     <!-- Brand -->
15     <a class="navbar-brand" th:href="@{/students}">Students Management</a>
16
17     <!-- Toggler/collapse Button -->
18     <button class="navbar-toggler" type="button" data-toggle="collapse"
19       data-target="#collapsibleNavbar">
20       <span class="navbar-toggler-icon"></span>
21     </button>
22
23     <!-- Navbar links -->
24     <div class="collapse navbar-collapse" id="collapsibleNavbar">
25       <ul class="navbar-nav">
26         <li class="nav-item"><a class="nav-link" th:href="@{/students}">Students
27           Management</a></li>
28       </ul>
29     </div>
30   </nav>
31
32   <br>
33   <br>
34   <div class="container">
35     <div class="row">
36       <div
37         class="col-lg-6 col-md col-sm-6 container justify-content-center card">
38         <div class="text-center">
39           <h1>Create New Students</h1>
40           <div class="card-body">
41             <form th:action="@{/students}" th:object="${student}"
42               method="POST">
43               <div class="form-group">
44                 <label>Student First Name</label> <input type="text"
45                   name="firstname" th:field="*{firstname}" class="form-control"
46                   placeholder="Enter Student First Name" />
47               </div>
48               <div class="form-group">
49                 <label>Student Last Name</label> <input type="text"
50                   name="lastname" th:field="*{lastname}" class="form-control"
51                   placeholder="Enter Student Last Name" />
52               </div>
53               <div class="form-group">
54                 <label>Student Email</label> <input type="text" name="email"
55                   th:field="*{email}" class="form-control"
56                   placeholder="Enter Student Email" />
57               </div>
58               <div class="form-group">
59                 <label>Student Score</label> <input type="text" name="score"
60                   th:field="*{score}" class="form-control"
61                   placeholder="Enter Student Score" />
62               </div>
63               <div class="box-footer">
64                 <button type="submit" class="btn btn-primary">Submit</button>
65               </div>
66             </form>
67           </div>
68         </div>
69       </div>
70     </div>
71   </body>
72 </html>
```

Update Data Student

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <meta charset="UTF-8">
5   <title>Student Management</title>
6   <link rel="stylesheet"
7     href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
8     integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
9     crossorigin="anonymous">
10 </head>
11 <body>
12
13   <nav class="navbar navbar-expand-md bg-dark navbar-dark">
14     <!-- Brand -->
15     <a class="navbar-brand" th:href="@{/students}">Students Management</a>
16
17     <!-- Toggler/collapsible Button -->
18     <button class="navbar-toggler" type="button" data-toggle="collapse"
19       data-target="#collapsibleNavbar">
20       <span class="navbar-toggler-icon"></span>
21     </button>
22
23     <!-- Navbar links -->
24     <div class="collapse navbar-collapse" id="collapsibleNavbar">
25       <ul class="navbar-nav">
26         <li class="nav-item"><a class="nav-link" th:href="@{/students}">Students
27           Management</a></li>
28       </ul>
29     </div>
30   </nav>
31   <br>
32   <br>
33   <div class="container">
34     <div class="row">
35       <div
36         class="col-lg-6 col-md col-sm-6 container justify-content-center card">
37         <div class="text-center">
38           <h1>Update Students</h1>
39         <div class="card-body">
40           <form th:action="@{/students (id=${student.id})}"
41             th:object="${student}" method="POST">
42             <div class="form-group">
43               <label>Student First Name</label> <input type="text"
44                 name="firstname" th:field="*{firstname}" class="form-control"
45                 placeholder="Enter Student First Name" />
46             </div>
47             <div class="form-group">
48               <label>Student Last Name</label> <input type="text"
49                 name="lastname" th:field="*{lastname}" class="form-control"
50                 placeholder="Enter Student Last Name" />
51             </div>
52             <div class="form-group">
53               <label>Student Email</label> <input type="text" name="email"
54                 th:field="*{email}" class="form-control"
55                 placeholder="Enter Student Email" />
56             </div>
57             <div class="form-group">
58               <label>Student Score</label> <input type="text" name="score"
59                 th:field="*{score}" class="form-control"
60                 placeholder="Enter Student Score" />
61             </div>
62             <div class="box-footer">
63               <button type="submit" class="btn btn-primary">Submit</button>
64             </div>
65           </form>
66         </div>
67       </div>
68     </div>
69   </div>
70 </body>
71 </html>
```

Databases with MySQL

Create and Connection

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/dbstudents?createDatabaseIfNotExist=true&allowPublicKeyRetrieval=true&useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=
4
5 spring.datasource.hikari.connection-timeout=20000
6 spring.datasource.hikari.minimum-idle=5
7 spring.datasource.hikari.maximum-pool-size=100
8 spring.datasource.hikari.idle-timeout=300000
9 spring.datasource.hikari.max-lifetime=1200000
10 spring.datasource.hikari.auto-commit=true
11
12 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
13 spring.jpa.properties.hibernate.id.new_generator_mappings=false
14 spring.jpa.properties.hibernate.format_sql=true
15
16 spring.jpa.hibernate.ddl-auto=update
17 spring.jpa.show-sql=true
18
19 spring.jpa.open-in-view=true
20
21 spring.servlet.multipart.max-file-size=15MB
22 spring.servlet.multipart.max-request-size=15MB
23
```

Table

<input type="checkbox"/>				2	luonverakcambo@gmail.com	Verak	Luon	90
<input type="checkbox"/>				3	sokcheata@gmail.com	Sok	Cheata	98
<input type="checkbox"/>				4	nhebrany@gmail.com	Nheb	Rany	97

Check all With selected:

Result

First Name	Last Name	Email	Score	Actions
Verak	Luon	luonverakcambo@gmail.com	90.0	<button>Update</button> <button>Delete</button>
Sok	Cheata	sokcheata@gmail.com	98.0	<button>Update</button> <button>Delete</button>
Nheb	Rany	nhebrany@gmail.com	97.0	<button>Update</button> <button>Delete</button>

When Click Button Add or Update

Create New Students

Student First Name
Verak

Student Last Name
Luon

Student Email
luonverakcambo@gmail.com

Student Score
98

Submit

What is API?

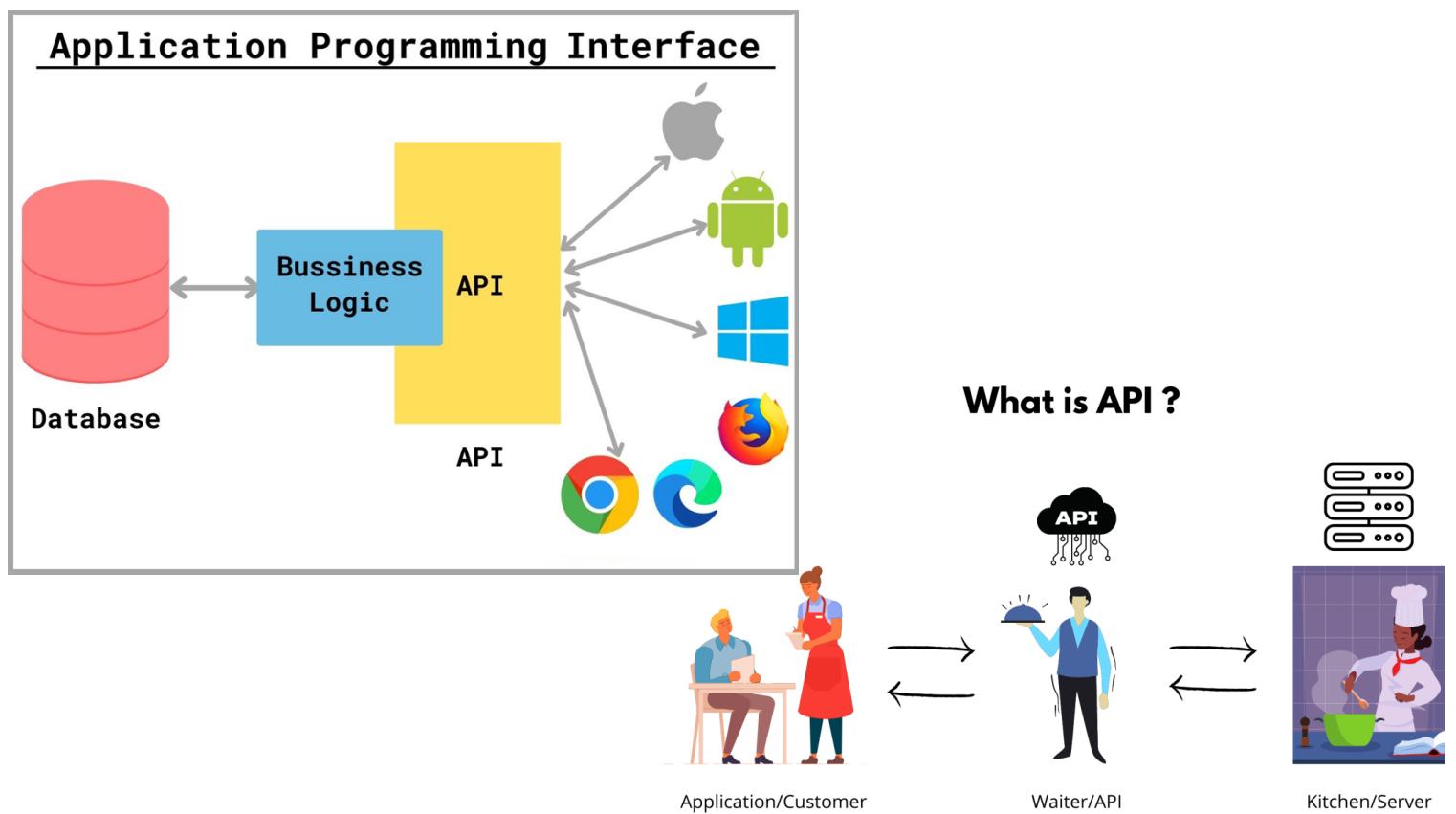
_API stand for Application Programming Interface

What is REST API?

_REST stand for Representation State Transfer. REST defines a set of function like GET, POST, PUT, DELETE, Etc. Clients can use to access server data. Clients and server exchange data use HTTP.

What is HTTP?

_HTTP stand for Hypertext Transfer Protocol



What is JSON?

_JSON stand for JavaScript Object Notation

Syntax:

```
{  
    "Key": "Value"  
    "Key":  
    [  
        {"Key": "Value"},  
        {"Key": "Value"},  
    ]  
}
```

For example

```
[{"book0":  
    {"title": "Java Programming All in 1",  
     "author": "Jeff Prosise",  
     "isbn": "0-7356-1376-1"}],  
 {"book1":  
    {"title": "Advance Java Cook Book",  
     "author": "Fergal Grimes",  
     "isbn": "1-930110-19-7"}]
```

JSON Structure

```
[  
  {  
    "books" :  
      {  
        "title" :  
          "Java Programming All in 1",  
        "author" :  
          "Jeff Prosise",  
        "isbn" :  
          "0-7356-1376-1"  
      }  
  }  
]
```

```
START_ARRAY  
  START_OBJECT  
    KEY_NAME  
      START_OBJECT  
        KEY_NAME  
          VALUE_STRING  
        KEY_NAME  
          VALUE_STRING  
        KEY_NAME  
          VALUE_STRING  
    END_OBJECT  
  END_OBJECT  
END_ARRAY
```

For Example:

```
-- 20<dependencies>
21    <dependency>
22        <groupId>org.projectlombok</groupId>
23        <artifactId>lombok</artifactId>
24        <version>1.18.24</version>
25        <scope>provided</scope>
26    </dependency>
27
28    <dependency>
29        <groupId>org.springframework.boot</groupId>
30        <artifactId>spring-boot-starter-data-jpa</artifactId>
31    </dependency>
32    <dependency>
33        <groupId>org.springframework.boot</groupId>
34        <artifactId>spring-boot-starter-web</artifactId>
35    </dependency>
36
37    <dependency>
38        <groupId>com.mysql</groupId>
39        <artifactId>mysql-connector-j</artifactId>
40        <scope>runtime</scope>
41    </dependency>
42    <dependency>
43        <groupId>org.springframework.boot</groupId>
44        <artifactId>spring-boot-starter-tomcat</artifactId>
45        <scope>provided</scope>
46    </dependency>
47    <dependency>
48        <groupId>org.springframework.boot</groupId>
49        <artifactId>spring-boot-starter-test</artifactId>
50        <scope>test</scope>
51    </dependency>
52</dependencies>
53
```

```

1 package com.example.demo.model;
2
3 import jakarta.persistence.Entity;
4
5 @Getter
6 @Setter
7 @NoArgsConstructor
8 @AllArgsConstructor
9 @Entity
10 @Table(name="tblpeople")
11 public class People {
12
13     @Id
14     @GeneratedValue(strategy = GenerationType.AUTO)
15     private long id;
16     private String name;
17     private String address;
18
19 }
20
21
22
23
24
25
26
27 }
28

```

```

1 package com.example.demo.service;
2
3 import java.util.List;
4
5 @Service
6 public class PeopleService {
7
8     @Autowired
9     private PeopleRepo peopleRepo;
10
11     public List<People> findAllpeople() {
12         return peopleRepo.findAll();
13     }
14
15     public People findById(long id) {
16         return peopleRepo.findById(id).orElse(new People());
17     }
18
19     public List<People> findByAddress(String address) {
20         return peopleRepo.findByAddress(address);
21     }
22
23     public void save(People people) {
24         peopleRepo.save(people);
25     }
26
27     public void update(People people) {
28         People pl=peopleRepo.findById(people.getId()).orElse(new People());
29         if(pl.getId()==0)
30             pl.setName("No People");
31         else
32             peopleRepo.save(people);
33     }
34
35     public void delete(People people) {
36         peopleRepo.deleteById(people.getId());
37     }
38
39 }
40
41
42
43
44
45 }
46

```

```

1 package com.example.demo.controller;
2
3 import java.util.List;
4
5 @RestController
6 @RequestMapping("/api/people")
7 public class MyController {
8
9     @Autowired
10     private PeopleService people;
11
12     @GetMapping("")
13     public List<People> listAllpeople() {
14         return people.findAllpeople();
15     }
16
17     @GetMapping("/id/{id}")
18     public People findByID(@PathVariable long id) {
19         return people.findById(id);
20     }
21
22     @GetMapping("/address/{address}")
23     public List<People> findByaddress(@PathVariable String address){
24         return people.findByAddress(address);
25     }
26
27 }
28
29
30
31
32
33
34
35
36
37 }
38

```

```

1 package com.example.demo.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @RequestMapping("/api/people")
7 public class MyController2 {
8
9     @Autowired
10     private PeopleService peopleService;
11
12     @PostMapping("")
13     public People add(@RequestBody People people) {
14         peopleService.save(people);
15         return people;
16     }
17
18     @PutMapping("")
19     public People update(@RequestBody People people) {
20         peopleService.update(people);
21         return people;
22     }
23
24     @DeleteMapping("")
25     public People delete(@RequestBody People people) {
26         peopleService.delete(people);
27         return people;
28     }
29
30 }
31
32
33
34
35
36
37

```

```
1 package com.example.demo.repo;
2
3 import java.util.List;
4
5 public interface PeopleRepo extends JpaRepository<People, Long>{
6     List<People> findByAddress(String address);
7 }
8
9
application.properties X
1 spring.datasource.url=jdbc:mysql://localhost:3306/springapi?createDatabaseIfNotExist=true&allowPublicKeyRetrieval=true&useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=
4
5 spring.datasource.hikari.connection-timeout=20000
6 spring.datasource.hikari.minimum-idle=5
7 spring.datasource.hikari.maximum-pool-size=100
8 spring.datasource.hikari.idle-timeout=300000
9 spring.datasource.hikari.max-lifetime=1200000
10 spring.datasource.hikari.auto-commit=true
11
12 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
13 spring.jpa.properties.hibernate.id.new_generator_mappings=false
14 spring.jpa.properties.hibernate.format_sql=true
15
16 spring.jpa.hibernate.ddl-auto=update
17 spring.jpa.show-sql=true
18
19 spring.jpa.open-in-view=true
```

			id	address	name				
		← T →							
<input type="checkbox"/>		Edit		Copy		Delete	1	Phnom Penh	Luon Verak
<input type="checkbox"/>		Edit		Copy		Delete	2	Phnom Penh	Heng Visal
<input type="checkbox"/>		Edit		Copy		Delete	3	Phnom Penh	Jonh Son
<input type="checkbox"/>		Edit		Copy		Delete	4	Angtasom	Seng Chan Lyna
<input type="checkbox"/>		Edit		Copy		Delete	5	Phnom Penh	Chomrern Sopheak
<input type="checkbox"/>		Edit		Copy		Delete	7	Takeo	Chhorn Chenda
<input type="checkbox"/>		Edit		Copy		Delete	102	Phnom Penh	Leng Dalin
<input type="checkbox"/>		Edit		Copy		Delete	152	Phnom Penh	So Da

We need Install Postman for testing API

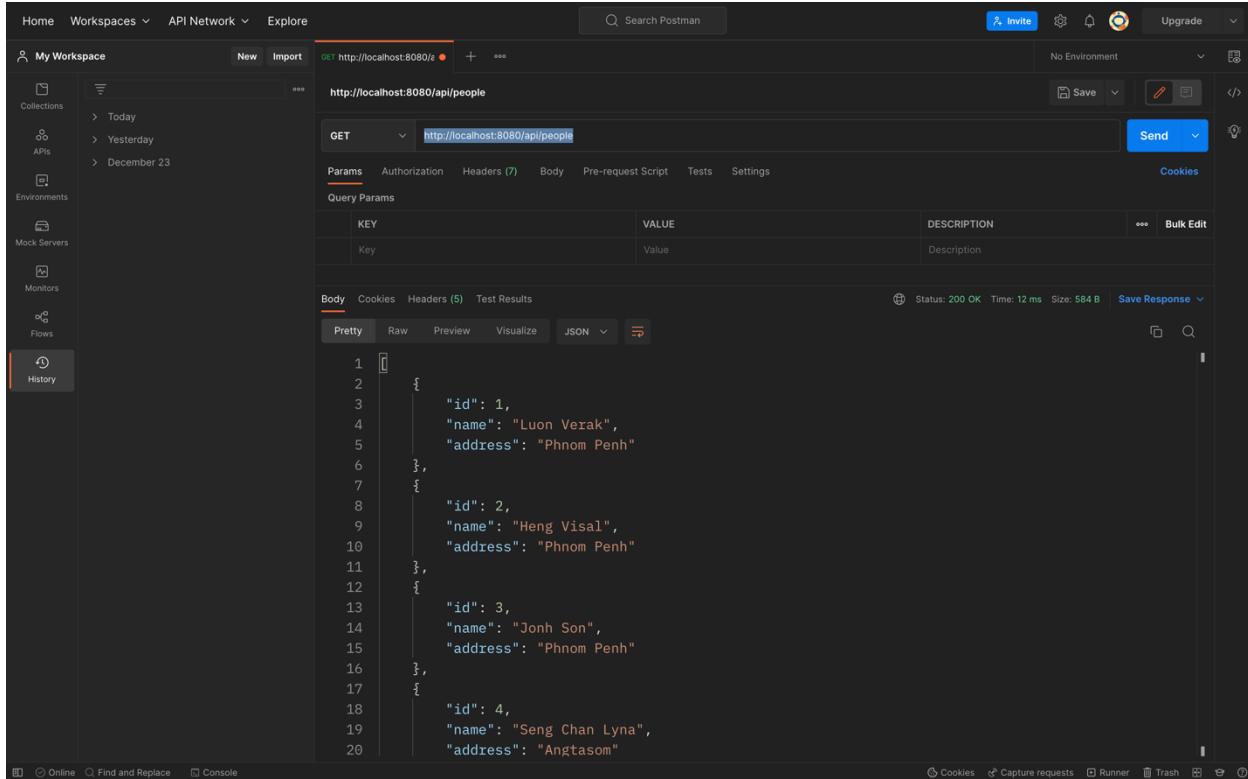
<https://www.postman.com/downloads/>



The image shows a screenshot of a web browser window displaying a JSON response from a local API endpoint. The URL in the address bar is `localhost:8080/api/people`. The JSON data lists several people with their IDs, names, and addresses:

```
[{"id": 1, "name": "Luon Verak", "address": "Phnom Penh"}, {"id": 2, "name": "Heng Visal", "address": "Phnom Penh"}, {"id": 3, "name": "Jonh Son", "address": "Phnom Penh"}, {"id": 4, "name": "Seng Chan Lyna", "address": "Angtasom"}, {"id": 5, "name": "Chomrern Sopheak", "address": "Phnom Penh"}, {"id": 7, "name": "Chhorn Chenda", "address": "Takeo"}, {"id": 102, "name": "Leng Dalin", "address": "Phnom Penh"}, {"id": 152, "name": "So Da", "address": "Phnom Penh"}]
```

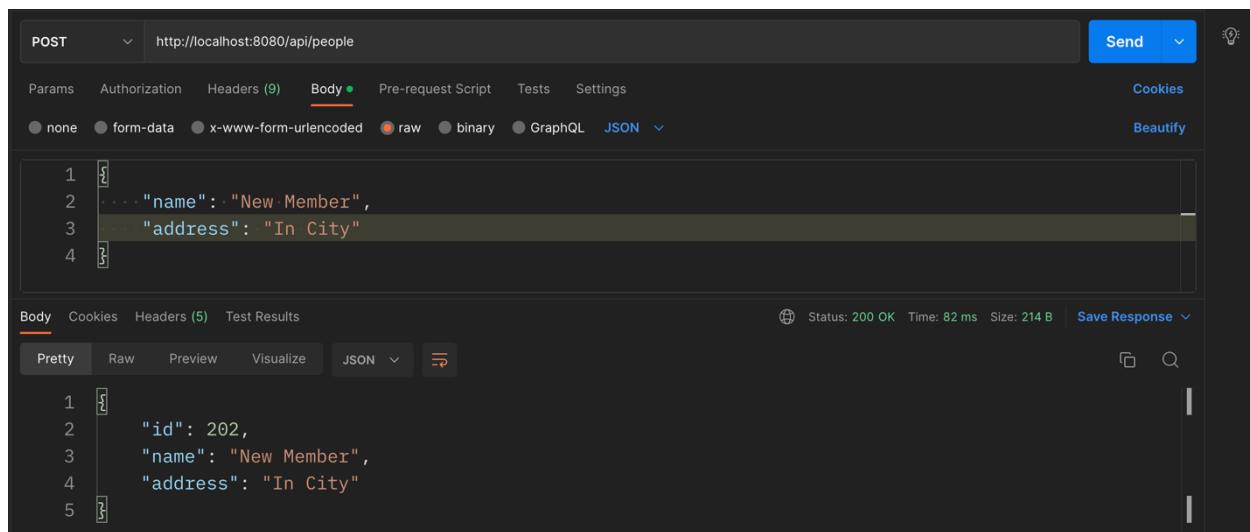
Get API in Postman សំដែរលើការទាញពិន្ទុយតាមរយៈEndpoint people



The screenshot shows the Postman interface with a successful GET request to `http://localhost:8080/api/people`. The response body is a JSON array containing four objects, each representing a person with fields `id`, `name`, and `address`.

```
1 [
2   {
3     "id": 1,
4     "name": "Luon Verak",
5     "address": "Phnom Penh"
6   },
7   {
8     "id": 2,
9     "name": "Heng Visal",
10    "address": "Phnom Penh"
11 },
12 {
13   "id": 3,
14   "name": "Jonth Son",
15   "address": "Phnom Penh"
16 },
17 {
18   "id": 4,
19   "name": "Seng Chan Lyna",
20   "address": "Angtasom"
21 ]
```

Post API in Postman សំដែរលើការបញ្ចប់ពិន្ទុយតាមរយៈEndpoint people

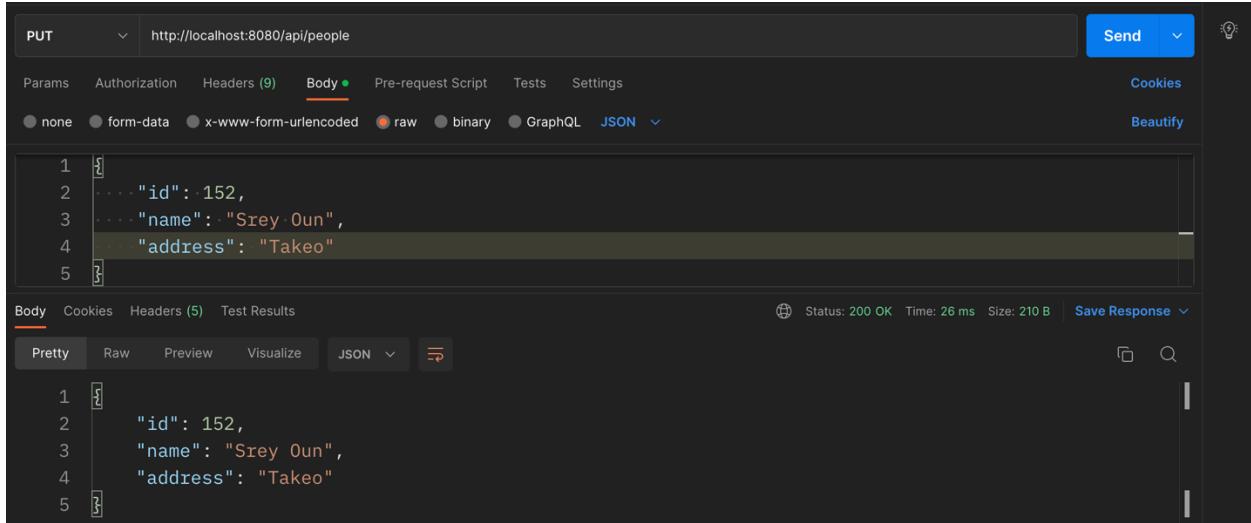


The screenshot shows the Postman interface with a successful POST request to `http://localhost:8080/api/people`. The request body is a JSON object with fields `name` and `address`. The response body is a JSON object with fields `id`, `name`, and `address`.

```
1 [
2   {
3     "name": "New Member",
4     "address": "In City"
5   }
]
```

```
1 {
2   "id": 202,
3   "name": "New Member",
4   "address": "In City"
5 }
```

Put API in Postman សំដែរលើការកែត្របន្ទីយតាមរយៈ Endpoint people

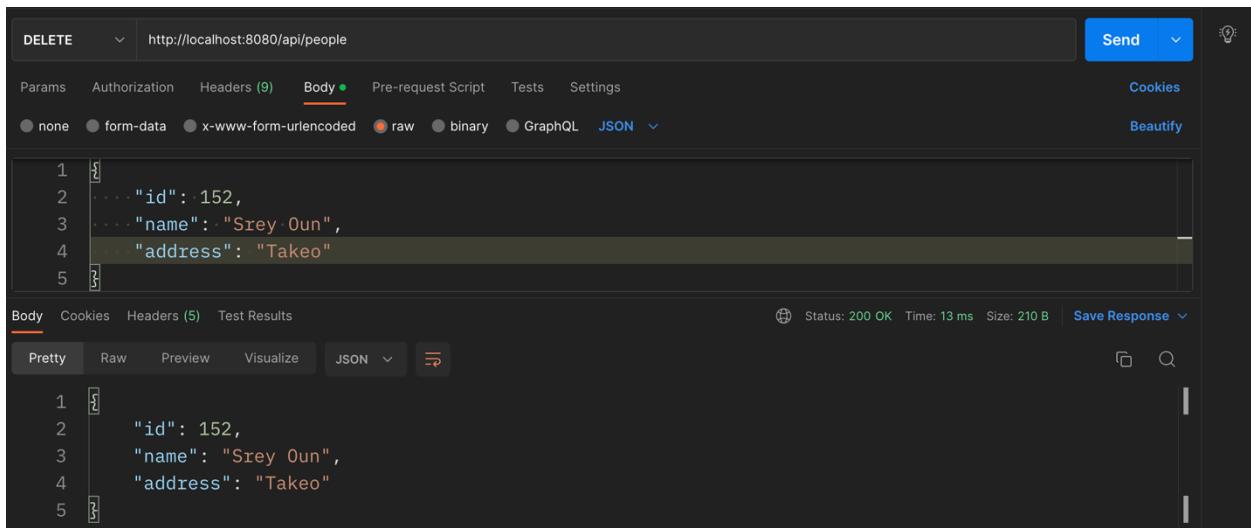


The screenshot shows the Postman interface with a PUT request to `http://localhost:8080/api/people`. The Body tab is selected, displaying a JSON object with fields `id`, `name`, and `address`. The response status is 200 OK.

```
1 {  
2   "id": 152,  
3   "name": "Srey Oun",  
4   "address": "Takeo"  
5 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 26 ms Size: 210 B Save Response

Delete API in Postman សំដែរលើការលុបត្របន្ទីយតាមរយៈ Endpoint people



The screenshot shows the Postman interface with a DELETE request to `http://localhost:8080/api/people`. The Body tab is selected, displaying a JSON object with fields `id`, `name`, and `address`. The response status is 200 OK.

```
1 {  
2   "id": 152,  
3   "name": "Srey Oun",  
4   "address": "Takeo"  
5 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 13 ms Size: 210 B Save Response

Restful Spring boot + React Js

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-tomcat</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Backend

```
1 package com.example.demo;
2
3 import org.springframework.beans.factory.annotation.Autowired;[]
10
11 @SpringBootApplication
12 public class Project1Application implements CommandLineRunner{
13
14     public static void main(String[] args) {
15         SpringApplication.run(Project1Application.class, args);
16     }
17
18     @Autowired
19     private EmployeeRepository employeeRepository;
20
21     @Override
22     public void run(String... args) throws Exception {
23         Employee employee=new Employee();
24         employee.setFirstName("Verak");
25         employee.setLastName("Luon");
26         employee.setEmail("luonverak@gmail.com");
27         employeeRepository.save(employee);
28     }
29
30 }
```

```
1 package com.example.demo.controller;
2
3 import java.util.List;[]
21
22 @CrossOrigin("http://localhost:3000/")
23 @RestController
24 @RequestMapping("/api/employee")
25 public class EmployeeController {
26
27     @Autowired
28     private EmployeeRepository employeeRepository;
29
30     @GetMapping
31     public List<Employee> getAllEmployee(){
32         return employeeRepository.findAll();
33     }
34     @PostMapping("")
35     public Employee createEmployee(@RequestBody Employee employee) {
36         return employeeRepository.save(employee);
37     }
}
```

```
1 package com.example.demo.exception;
2
3+import org.springframework.http.HttpStatus;□
4
5 @ResponseStatus(value = HttpStatus.NOT_FOUND)
6 public class ResourceNotFoundException extends RuntimeException{
7     public ResourceNotFoundException(String message) {
8         super(message);
9     }
10 }
11
12 }
```

```
EmployeeRepository.java ×
1 package com.example.demo.repository;
2
3+import org.springframework.data.jpa.repository.JpaRepository;□
4
5 public interface EmployeeRepository extends JpaRepository<Employee, Long>{
6
7 }
10
1 package com.example.demo.model;
2
3+import jakarta.persistence.Column;□
13
14 @Getter
15 @Setter
16 @NoArgsConstructor
17 @AllArgsConstructor
18 @Entity
```

Frontend React Js

Command for install project

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    GITLENS

● macbook@Macbooks-MacBook-Pro React % node -v
  v18.12.1
● macbook@Macbooks-MacBook-Pro React % npm -v
  8.19.2
○ macbook@Macbooks-MacBook-Pro React % npx create-react-app react-hook-frontend
Creating a new React app in /Users/macbook/Desktop/React/react-hook-frontend.
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    GITLENS

● macbook@Macbooks-MacBook-Pro React % npm install bootstrap --save
added 2 packages, and audited 3 packages in 1s
2 packages are looking for funding
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• macbook@Macbooks-MacBook-Pro React % npm install axios --save
added 9 packages, and audited 12 packages in 3s
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
○ macbook@Macbooks-MacBook-Pro React % □

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• macbook@Macbooks-MacBook-Pro React % npm install react-router-dom --save
added 8 packages, and audited 20 packages in 5s
3 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
○ macbook@Macbooks-MacBook-Pro React % □
```

```
App.js M
react-hook-frontend > src > App.js > App
You, 2 seconds ago | 1 author (You)
1 import './App.css';
2 import { BrowserRouter , Route,Routes} from 'react-router-dom';
3 import Footer from './components/Footer';
4 import HeaderComponents from './components/HeaderComponents';
5 import ListEmployeeComponents from './components/ListEmployeeComponents';
6 import AddEmployee from './components/AddEmployee';
7
8 function App() {
9   return [
10     <div>
11       <HeaderComponents/>
12       <BrowserRouter>
```

App.css M X

react-hook-frontend > src > App.css > .footer

You, 1 second ago | 1 author (You)

```
1 .footer{  
2   position: absolute;  
3   bottom: 0;  
4   width: 100%;  
5   height: 50px;      You, 1 second ago • Uncommitted changes  
6   background-color: red;  
7   text-align: center;  
8   color: white;  
9 }
```

EmployeeService.js U X

react-hook-frontend > src > service > EmployeeService.js > EmployeeService > deleteEmployee

```
1 import axios from 'axios';  
2  
3 const EMPLOYEE_BASE_REST_API_URL ="http://localhost:8080/api/employee";  
4  
5 class EmployeeService{  
6  
7   getAllEmployee(){  
8     return axios.get(EMPLOYEE_BASE_REST_API_URL)  
9   }  
10  createEmployee(employee){  
11    return axios.post(EMPLOYEE_BASE_REST_API_URL,employee)  
12  }  
13  getEmployeeByID(employeeId){
```

```
js AddEmployee.js 1, U ×
react-hook-frontend > src > components > AddEmployee.js > [o] AddEmployee
1 import React, { useEffect, useState } from 'react'
2 import EmployeeService from '../service/EmployeeService'
3 import {Link, useNavigate, useParams} from 'react-router-dom';
4 const AddEmployee = () => {
5
6     const [firstName, setFirstName]=useState('')
7     const [lastName, setLastName]=useState('')
8     const [email, setEmail]=useState('')
9
10    const navigate = useNavigate();
11    const {id}=useParams();
12
13    const saveOrUpdateEmployee=(e) =>{
14        e.preventDefault();
15        const employee={firstName,lastName,email}
16
17        if(id){
18            EmployeeService.updateEmployee(id,employee).then((respone)=>{
19                navigate('/employee')
20            }).catch(error => {
21                console.log(error);
22            })
23        }else{
24            EmployeeService.createEmployee(employee).then((respone)=>{
25
26                console.log(respone.data);
27                navigate('/employee');
28            }).catch(error => {
29                console.log(error);
30            })
31        }
32
33    }
34
}

```

```
js AddEmployee.js 1, U ×
react-hook-frontend > src > components > AddEmployee.js > [o] AddEmployee
35 useEffect(()=>{
36     EmployeeService.getEmployeeByID(id).then((respone)=>{
37         setFirstName(respone.data.firstName)
38         setLastName(respone.data.lastName)
39         setEmail(respone.data.email)
40     }).catch(error=>{
41         console.log(error);
42     })
43 },[ ])
44
45 const title = ()=>{
46     if(id){
47         return <h2 className='text-center'>Update Information Employee</h2>
48     }
49 }

```

A screenshot of a code editor displaying the `AddEmployee.js` file. The code is written in JSX (JavaScript with XML-like structures). It defines a component structure with nested `div`, `form`, and `input` elements. The `input` fields have placeholder text and onChange event handlers that update state variables `firstName` and `lastName`. The code editor interface includes a navigation bar at the top, a vertical scrollbar on the right, and a status bar at the bottom indicating the current line (Ln 107), column (Col 11), and encoding (UTF-8).

```
53 return (
54     <div>
55         <div className='container'>
56             <div className='row'>
57
58                 <div className='card col-md-6 offset-md-3'>
59                     <br></br>
60                     <br></br>
61                     {
62                         title()
63                     }
64                     <div className='card-body'>
65                         <form>
66                             <div className='form-group mb-2'>
67                                 <label className='form-label'> First Name </label>
68                                 <input
69                                     type="text"
70                                     placeholder='Enter First Name'
71                                     className='form-control'
72                                     value={firstName}
73                                     onChange={(e) => setFirstName(e.target.value)}
74
75
76                                 </input>
77                             </div>
78                             <div className='form-group mb-2'>
79                                 <label className='form-label'> Last Name </label>
80                                 <input
81                                     type="text"
82                                     placeholder='Enter Last Name'
83                                     className='form-control'
84                                     value={lastName}
85                                     onChange={(e) => setLastName(e.target.value)}
86
87                                 </input>
88                         </div>
89                     </div>
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
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
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
878
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
978
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2
```

`Footer.js`

```
react-hook-frontend > src > components > Footer.js > Footer
1 import React from 'react'
2
3 function Footer() {
4   return (
5     <div>
6       <footer>
7         <nav className='footer'>
8           <span>All Right Spring Boot Develop on 23 December 2022 by Luon Verak</span>
9         </nav>
10      </footer>
11    </div>
12  )
13}
14
15 export default Footer
```

`HeaderComponents.js`

```
react-hook-frontend > src > components > HeaderComponents.js > HeaderComponents
1 import React from 'react'
2
3 function HeaderComponents() {
4   return (
5     <div>
6       <header>
7         <nav className='navbar navbar-expand-md navbar-dark bg-dark'>
8           <div>
9             <a href='https://www.facebook.com/?_rdc=2&_rdr' className='navbar-brand'>
10               Employee Management System
11             </a>
12           </div>
13         </nav>
14       </header>
```

```
File ListEmployeeComponents.js X
react-hook-frontend > src > components > ListEmployeeComponents.js > ListEmployeeComponents
1 import React, { useEffect, useState } from 'react'
2 import { Link } from 'react-router-dom';
3 import EmployeeService from '../service/EmployeeService';
4
5 function ListEmployeeComponents() {
6     const [employee, setEmployee] = useState([]);
7     useEffect(() => {
8         getAllEmployee();
9     }, [])
10
11     const getAllEmployee=()=>{ 
12         EmployeeService.getAllEmployee().then((response) => {
13             setEmployee(response.data)
14             console.log(response.data)
15         }).catch(
16             error => {
17                 console.log(error);
18             }
19         )
20     }
21     const deleteEmployee = (employeeId) => {
22         EmployeeService.deleteEmployee(employeeId).then((response)=>{
23             getAllEmployee();
24         }).catch(error =>{
25             console.log(error);
26         })
27     }
28     return (
29         <div className='container'>
30             <h2 className='text-center'>Information of Employee</h2>
31             <Link to="/add-employee" className="btn btn-primary mb-2"> Add Employee</Link>
32             <br><br>
33             <table className='table table-bordered table-striped'>
34                 <thead>
35                     <th>Employee-ID</th>
36                     <th>FirstName</th>
37                     <th>LastName</th>
38                     <th>E-mail</th>
39                     <th>Actions</th>
40                 </thead>
41                 <tbody>
```

```
File ListEmployeeComponents.js X
react-hook-frontend > src > components > ListEmployeeComponents.js > ListEmployeeComponents
28     return []
29     <div className='container'>
30         <h2 className='text-center'>Information of Employee</h2>
31         <Link to="/add-employee" className="btn btn-primary mb-2"> Add Employee</Link>
32         <br><br>
33         <table className='table table-bordered table-striped'>
34             <thead>
35                 <th>Employee-ID</th>
36                 <th>FirstName</th>
37                 <th>LastName</th>
38                 <th>E-mail</th>
39                 <th>Actions</th>
40             </thead>
41             <tbody>
```

Table Databases

			id	email	first_name	last_name
<input type="checkbox"/>	 Edit	 Copy	 Delete	1 luonverak@gmail.com	Verak	Luon
<input type="checkbox"/>	 Edit	 Copy	 Delete	2 hengvisal@gmail.com	Heng	Visal
<input type="checkbox"/>	 Edit	 Copy	 Delete	4 snegdalin@gmail.com	Seng	Dalin

 Check all With selected:  Edit  Copy  Delete  Export

Result API

```
http://localhost:8080/api/employee
localhost:8080/api/employee

[  
  {  
    id: 1,  
    firstName: "Verak",  
    lastName: "Luon",  
    email: "luonverak@gmail.com"  
  },  
  {  
    id: 2,  
    firstName: "Heng",  
    lastName: "Visal",  
    email: "hengvisal@gmail.com"  
  },  
  {  
    id: 4,  
    <root>[1].lastName  
    firstName: "Seng",  
    lastName: "Dalin",  
    email: "snegdalin@gmail.com"  
  }  
]
```

React App

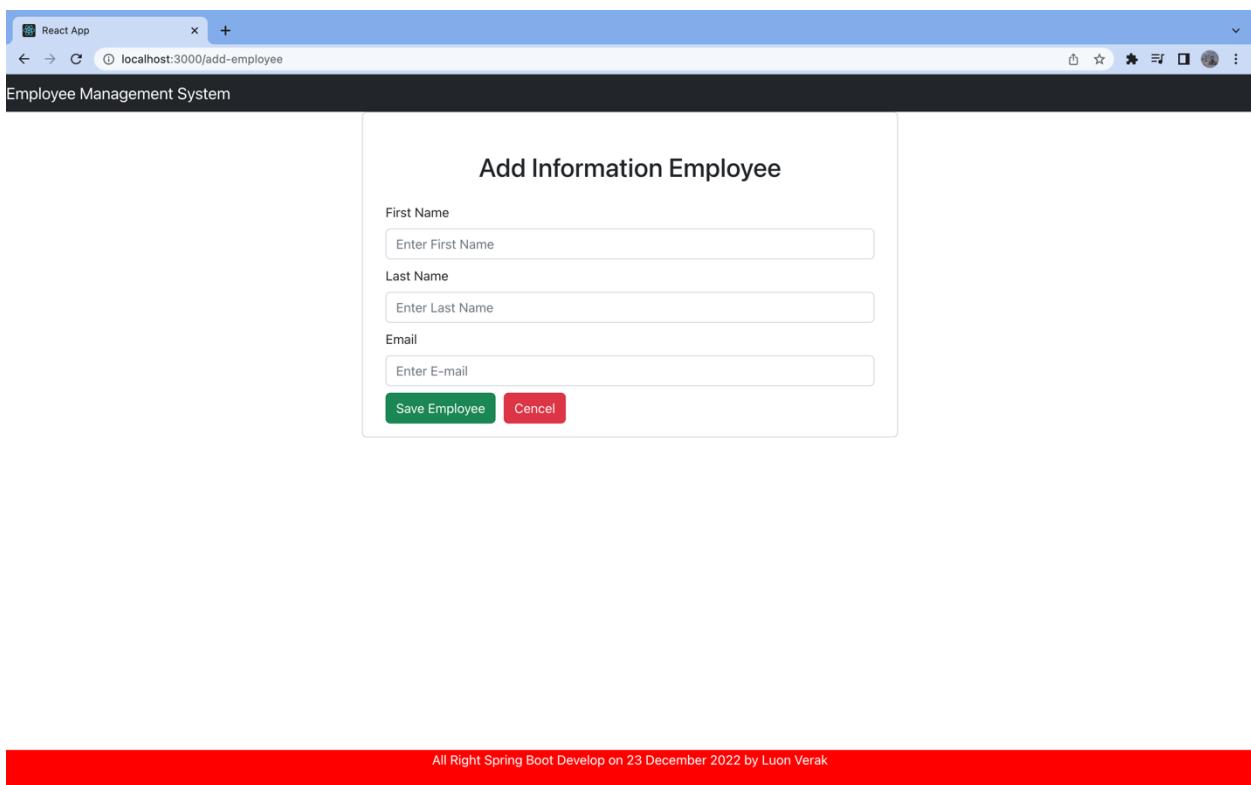
localhost:3000

Employee Management System

Information of Employee

Information of Employee						
Add Employee		Employee-ID	FirstName	LastName	E-mail	Actions
1	Verak	Luon	luonverak@gmail.com	<button>Update</button>	<button>Delete</button>	
2	Heng	Visal	hengvisal@gmail.com	<button>Update</button>	<button>Delete</button>	
4	Seng	Dalin	snegdalin@gmail.com	<button>Update</button>	<button>Delete</button>	

Button Add New Employee or Update



A screenshot of a web browser window titled "React App". The address bar shows "localhost:3000/add-employee". The main content area is titled "Add Information Employee". It contains three input fields: "First Name" (placeholder "Enter First Name"), "Last Name" (placeholder "Enter Last Name"), and "Email" (placeholder "Enter E-mail"). Below the inputs are two buttons: a green "Save Employee" button and a red "Cancel" button. The browser's title bar says "Employee Management System". At the bottom of the page is a red footer bar with the text "All Right Spring Boot Develop on 23 December 2022 by Luon Verak".

React App

localhost:3000/add-employee

Employee Management System

Add Information Employee

First Name
Enter First Name

Last Name
Enter Last Name

Email
Enter E-mail

Save Employee Cancel

All Right Spring Boot Develop on 23 December 2022 by Luon Verak