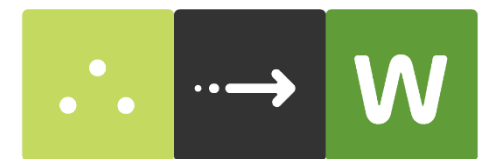




Woodland
Academy

1.8 PersonProject 中級

- UserBlogController追加の設定
- ユーザーのブロッガー一覧画面の作成
- ブログ詳細画面の作成
- カテゴリ別の記事一覧画面の作成



Shape Your Future

目次

- 1 **UserBlogController
追加の設定**
- 2 ユーザーのブロッ
グ一覧の画面作成
- 3 Blog詳細画面の作成
- 4 カテゴリ別の記事一
覧画面の作成

UserBlogControllerクラスの作成①

- UserBlogControllerクラスを作成し、下記のソースを書き写してください。ソースの解説はコメントをみて理解を進めてください。

```
UserBlogController.java ×
1 package blog.example.controller;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RequestMapping;
11
12 import blog.example.model.entity.BlogEntity;
13 import blog.example.model.entity.CategoryEntity;
14 import blog.example.service.BlogService;
15 import blog.example.service.CategoryService;
16
17
```

```
18 @Controller
19 @RequestMapping("/blog")
20 public class UserBlogController {
21     /**
22      * blogテーブルを操作するための
23      * Serviceクラス
24      */
25     @Autowired
26     BlogService blogService;
27     /**
28      * categoryテーブルを操作するための
29      * Serviceクラス
30      */
31     @Autowired
32     CategoryService categoryService;
33
34     /**
35      * blogテーブルから全てのブログ記事を取得
36      * categoryテーブルから全てのカテゴリを取得
37      * 取得した情報をセットして画面から参照可能にします。
38      */

```


UserBlogControllerクラスの作成②

- UserBlogControllerクラスを作成し、下記のソースを書き写してください。ソースの解説はコメントをみて理解を進めてください。

```

34  /**
35   * blogテーブルから全てのブログ記事を取得
36   * categoryテーブルから全てのカテゴリを取得
37   * 取得した情報をセットして画面から参照可能にします。
38   */
39   //ブログ画面の表示(ログイン操作無し)
40  @GetMapping("/")
41  public String getBlogUserPage(Model model) {
42      //ブログの情報を全て取得する
43      List<BlogEntity>blogList = blogService.selectAll();
44      model.addAttribute("blogList",blogList);
45      //カテゴリの情報を全て取得する
46      List<CategoryEntity>categoryList = categoryServie.findAll();
47      model.addAttribute("categoryList",categoryList);
48      return "blog.html";
49  }

```

UserBlogControllerクラスの作成③

- UserBlogControllerクラスを作成し、下記のソースを書き写してください。ソースの解説はコメントをみて理解を進めてください。

```

50  /**
51   * ブログ記事詳細画面を表示するための処理です。
52   * blogIdに紐づく内容をblogテーブルから取得
53   * categoryテーブルから全てのカテゴリを取得
54   * 取得した情報をセットして画面から参照可能にします。
55   */
56  @GetMapping("/detail/{blogId}")
57  public String getBlogUserDetailPage(@PathVariable Long blogId, Model model) {
58      //blogのテーブルの中から、blogIdで検索をかけて該当する該当するブログの情報を引っ張り出す。
59      BlogEntity blogs = blogService.selectByBlogId(blogId);
60      //カテゴリの情報を全て取得する
61      List<CategoryEntity>categoryList = categoryServie.findAll();
62      //blogs(ブログ記事詳細情報)とcategoryList(カテゴリ一覧)をmodelにセットし
63      //blog_detail.htmlから参照可能にする。
64      model.addAttribute("categoryList",categoryList);
65      model.addAttribute("blogs",blogs);
66      return "blog_detail.html";
67  }

```

UserBlogControllerクラスの作成④

- UserBlogControllerクラスを作成し、下記のソースを書き写してください。ソースの解説はコメントをみて理解を進めてください。

```

68  /**
69   * ブログの記事をカテゴリ別に表示するための処理
70   * categoryNameに紐づく内容をblogテーブルから取得
71   * categoryテーブルから全てのカテゴリを取得
72   * 取得した情報をセットして画面から参照可能にします。
73   */
74  @GetMapping("/category/list/{categoryName}")
75  public String getBlogCategoryAllPage(@PathVariable String categoryName, Model model) {
76      //blogのテーブルの中から、categoryNameで検索をかけて該当する該当するブログの情報を引っ張り出す。
77      List<BlogEntity> blogList = blogService.selectByCategoryName(categoryName);
78      //カテゴリの情報を全て取得する
79      List<CategoryEntity> categoryList = categoryService.findAll();
80      model.addAttribute("categoryList", categoryList);
81      model.addAttribute("categoryName", categoryName);
82      model.addAttribute("blogList", blogList);
83      //blogList(ブログ記事一覧)とcategoryList(カテゴリー一覧)とcategoryName(カテゴリー名)をmodelにセットし
84      //blog_category_list.htmlから参照可能にする。
85      return "blog_category_list.html";
86  }
87 }

```


目次

- 1 UserBlogController
追加の設定
- 2 ユーザーのブロッ
グ一覧の画面作成
- 3 Blog詳細画面の作成
- 4 カテゴリ別の記事一
覧画面の作成

blogの作成①

- blog.htmlを作成し、下記のソースを書き写してください。

```
blog.html ×
1<html lang="jp">
2
3<head>
4  <meta charset="UTF-8" />
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <link rel="preconnect" href="https://fonts.gstatic.com">
7  <link href="https://fonts.googleapis.com/css2?family=Indie+Flower&display=swap" rel="stylesheet">
8  <link rel="stylesheet" th:href="@{/css/reset.css}">
9  <link rel="stylesheet" th:href="@{/css/blog_style.css}">
10 <link rel="stylesheet" th:href="@{/css/pc_style.css}">
11</head>
12
13<body>
14  <header class="media__header">
15    <div class="header__logo">
16      <a th:href="@{/blog/}">
17        
18      </a>
19    </div>
20    <nav class="header__nav pc">
21      <ul class="flex__item">
22        <li class="nav__list" th:each="category : ${categoryList}">
23          <a th:href="@{/blog/category/list/{categoryName}(categoryName=${category.categoryName})}"
24            class="nav__link" th:text="${category.categoryName}"></a>
25        </li>
26      </ul>
27    </nav>
28  </header>
```


blogの作成②

- blog.htmlを作成し、下記のソースを書き写してください。

```

29<main>
30
31<div class="media-firstview">
32  <h1 class="media__fv-ttl">Commit to the growth<br>for everyone</h1>
33</div>
34
35<div class="container">
36  <p class="newpost__sub-ttl">New Post</p>
37  <h2 class="newpost__ttl">記事一覧</h2>
38  <div class="flex__item">
39    <div class="article-wrap">
40      <div class="flex__item">
41        <a class="article__card" th:href="@{/blog/detail/{blogId}(blogId=${blog.blogId})}"
42          th:each="blog:${blogList}">
43          <div class="article__card-eyecatch">
44            
45          </div>
46          <span class="article__card-cat" th:text=" ${blog.categoryName}">
47          </span>
48
49          <div class="article__card-content">
50            <h3 class="article__card-content-ttl" th:text=" ${blog.blogTitle}"></h3>
51          </div>
52        </a>
53      </div>
54    </div>

```

blogの作成③

- blog.htmlを作成し、下記のソースを書き写してください。

```

55<         <aside class="sidebar">
56<             <div class="category">
57<                 <h2 class="category__ttl">Category</h2>
58<                 <ul>
59<                     <li class="category__list" th:each="category : ${categoryList}"><a
60<                         th:href="@{/blog/category/list/{categoryName}(categoryName=${category.categoryName})}"
61<                         class="category__link" th:text="${category.categoryName}"></a></li>
62<                 </ul>
63<             </div>
64<         </aside>
65<     </div>
66< </div>
67< </main>
68< <footer class="footer">
69<     <p>Copyright © 2022 izawa-Sample All Rights Reserved.</p>
70< </footer>
71< </body>
72
73< </html>

```

目次

- 1 UserBlogController
追加の設定
- 2 ユーザーのブロッ
グ一覧の画面作成
- 3 Blog詳細画面の作成
- 4 カテゴリ別の記事一
覧画面の作成

blog_detailの作成①

- blog_detail.htmlを作成し、下記のソースを書き写してください。

```
blog_detail.html ×
1 <html lang="jp">
2
3 <head>
4   <meta charset="UTF-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="preconnect" href="https://fonts.gstatic.com">
7   <link href="https://fonts.googleapis.com/css2?family=Indie+Flower&display=swap" rel="stylesheet">
8   <link rel="stylesheet" th:href="@{/css/reset.css}">
9   <link rel="stylesheet" th:href="@{/css/blog_style.css}">
10  <link rel="stylesheet" th:href="@{/css/pc_style.css}">
11 </head>
12
```

blog_detailの作成②

- blog_detail.htmlを作成し、下記のソースを書き写してください。

```

13<body>
14  <header class="media_header">
15    <div class="header_logo">
16      <a th:href="@{/blog/}">
17        
18      </a>
19    </div>
20    <nav class="header_nav pc">
21      <ul class="flex_item">
22        <li class="nav_list" th:each="category : ${categoryList}">
23          <a th:href="@{/blog/category/list/{categoryName}(categoryName=${category.categoryName})}" class="nav_link" th:text="${category.categoryName}"></a>
24        </li>
25      </ul>
26    </nav>
27  </header>

```

blog_detailの作成③

- blog_detail.htmlを作成し、下記のソースを書き写してください。

```

28<main>
29    <div class="single__container">
30        <div class="article__header">
31            <span class="single__cat" th:text="${blogs.categoryName}"></span>
32            <h1 th:text="${blogs.blogTitle}"></h1>
33        </div>
34        <div class="article__eyecatch">
35            
36        </div>
37
38        <div class="article__content">
39            <p th:text="${blogs.message}"></p>
40        </div>
41    </div>
42</main>
43<footer class="footer">
44    <p>Copyright © 2022 izawa-Sample All Rights Reserved.</p>
45</footer>
46</body>
47
48</html>

```


目次

- 1 UserBlogController
追加の設定
- 2 ユーザーのブローグー
覧の画面作成
- 3 Blog詳細画面の作成
- 4 カテゴリ別の記事一
覧画面の作成

blog_category_listの作成①

- blog_category_list.htmlを作成し、下記のソースを書き写してください。

```
blog_category_list.html ×
1<html lang="jp">
2
3<head>
4  <meta charset="UTF-8" />
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <link rel="preconnect" href="https://fonts.gstatic.com">
7  <link href="https://fonts.googleapis.com/css2?family=Indie+Flower&display=swap" rel="stylesheet">
8  <link rel="stylesheet" th:href="@{/css/reset.css}">
9  <link rel="stylesheet" th:href="@{/css/blog_style.css}">
10 <link rel="stylesheet" th:href="@{/css/pc_style.css}">
11 </head>
12
```

blog_category_listの作成②

- blog_category_list.htmlを作成し、下記のソースを書き写してください。

```

12
13<body>
14  <header class="media_header">
15    <div class="header_logo">
16      <a th:href="@{/blog/}">
17        
18      </a>
19    </div>
20    <nav class="header_nav pc">
21      <ul class="flex_item">
22        <li class="nav_list" th:each="category : ${categoryList}">
23          <a th:href="@{/blog/category/list/{categoryName}{categoryName=${category.categoryName}}}" class="nav_link" th:text="${category.categoryName}"></a>
24        </li>
25      </ul>
26    </nav>
27  </header>
28  <main>
29

```


blog_category_listの作成③

- blog_category_list.htmlを作成し、下記のソースを書き写してください。

```

30
31<div class="media-firstview">
32    <h1 class="media__fv-ttl" th:text="${categoryName}"></h1>
33</div>
34
35<div class="container">
36    <p class="newpost__sub-ttl">New Post</p>
37    <h2 class="newpost__ttl">記事一覧</h2>
38    <div class="flex__item">
39        <div class="article-wrap">
40            <div class="flex__item">
41                <a class="article__card" th:href="@{/blog/detail/{blogId}(blogId=${blog.blogId})}"
42                    th:each="blog:${blogList}">
43                    <div class="article__card-eyecatch">
44                        
45                    </div>
46                    <span class="article__card-cat" th:text="${blog.categoryName}">
47                    </span>
48
49                    <div class="article__card-content">
50                        <h3 class="article__card-content-ttl" th:text="${blog.blogTitle}"></h3>
51                    </div>
52                </a>
53            </div>
54        </div>

```

blog_category_listの作成④

- blog_category_list.htmlを作成し、下記のソースを書き写してください。

```

55<aside class="sidebar">
56  <div class="category">
57    <h2 class="category__ttl">Category</h2>
58    <ul>
59      <li class="category__list" th:each="category : ${categoryList}">
60        <a th:href="@{/blog/category/list/{categoryName}(categoryName=${category.categoryName})}"
61          class="category__link" th:text="${category.categoryName}">
62          </a>
63        </li>
64      </ul>
65    </div>
66  </aside>
67</div>
68</div>
69</main>
70<footer class="footer">
71  <p>Copyright © 2022 izawa-Sample All Rights Reserved.</p>
72</footer>
73</body>
74
75</html>

```

cssの設定①

- blog_style.cssを作成し、下記のソースを書き写してください。

```
blog_style.css ×
1 @charset "utf-8";
2
3
4 img {
5   width: 100%;
6   height: auto;
7 }
8
9 a {
10  color: #707070;
11  text-decoration: none;
12 }
13
14 li {
15  list-style: none;
16 }
17
18 body {
19  color: #707070;
20  line-height: 1.7;
21  background: #f5f5f5;
22 }
23
24 .pc {
25  display: none;
26 }
27 .container {
28  width: 90%;
29  margin: 70px auto 0;
30  padding-bottom: 70px;
31 }
32
33 .flex_item {
34  display: flex;
35  justify-content: space-between;
36  flex-wrap: wrap;
37 }
```



```

38
39
40 .common__ttl {
41   font-size: 24px;
42   font-weight: bold;
43   position: relative;
44 }
45
46 .common__sub-ttl {
47   font-size: 16px;
48   font-family: "Indie Flower", cursive;
49   color: #b31566;
50 }
51
52 .media_header {
53   background: #fff;
54   padding: 20px 0;
55 }
56
57 .header__logo {
58   width: 180px;
59   margin: 0 auto;
60 }
61
```

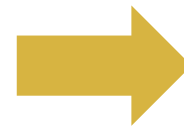

cssの設定②

- blog_style.cssを作成し、下記のソースを書き写してください。

```

61
62 /*-----blog----- */
63
64 .article-wrap {
65     padding-bottom: 40px;
66     border-bottom: 1px solid #c7c7c7;
67 }
68
69 .newpost__ttl {
70     font-size: 20px;
71     font-weight: bold;
72     position: relative;
73 }
74
75 .newpost__sub-ttl {
76     font-size: 1.2rem;
77     font-family: "Indie Flower", cursive;
78     color: #b31566;
79     font-weight: bold;
80 }
81
82 .media-firstview {
83     height: 560px;
84     background: url(../images/background.jpg) no-repeat;
85     background-size: cover;
86     background-position: center;
87     position: relative;
88     width: 100%;
89 }
90

```



```

90
91 .media__fv-ttl {
92     position: absolute;
93     top: 50%;
94     left: 50%;
95     transform: translate(-50%, -50%);
96     -webkit-transform: translate(-50%, -50%);
97     -moz-transform: translate(-50%, -50%);
98     -ms-transform: translate(-50%, -50%);
99     -o-transform: translate(-50%, -50%);
100     font-size: 24px;
101     text-align: center;
102     color: #594A4E;
103     width: 80%;
104     font-family: "Indie Flower", cursive;
105 }
106
107 .article__wrap {
108     margin-top: 30px;
109 }
110
111 .article__card {
112     width: 100%;
113     margin: 30px auto 0;
114     position: relative;
115     display: block;
116     background: #fff;
117     box-shadow: 0 3px 6px #f5f5f5;
118     border-radius: 15px;
119 }
120

```

cssの設定③

- blog_style.cssを作成し、下記のソースを書き写してください。

```

121 .article__card-eyecatch {
122     overflow: hidden;
123 }
124
125 .article__card-eyecatch img {
126     border-radius: 15px 15px 0 0;
127     object-fit: cover;
128     height: 150px;
129 }
130
131 .article__card-cat {
132     position: absolute;
133     top: 0;
134     left: 0;
135     background: #b31566;
136     color: #fff;
137     border-radius: 15px 0 0 0;
138     padding: 5px 20px;
139 }
140
141 .article__card-content {
142     padding: 20px;
143     margin-top: -7px;
144 }
145
146 .article__card-content-ttl {
147     margin-bottom: 20px;
148 }
149
150 .article__card-tag:first-child {
151     margin-left: 0;
152 }
153

```



```

154 .article__card-content-list {
155     position: absolute;
156     bottom: 5px;
157 }
158
159 .article__card-tag {
160     font-size: 12px;
161     margin-left: 15px;
162     display: inline-block;
163 }
164
165 .blog__link {
166     display: block;
167     margin: 30px auto 0;
168     width: 200px;
169     text-align: center;
170     padding: 15px 0;
171     background: #fff;
172     border: 2px solid #b31566;
173     border-radius: 30px;
174     color: #b31566;
175     transition: 0.4s;
176 }
177
178 .blog__link:hover {
179     background-color: #b31566;
180     color: #fff;
181     transition: 0.4s;
182 }
183

```

cssの設定④

- blog_style.cssを作成し、下記のソースを書き写してください。

```

184 /*-----sidebar----- */
185
186 .sidebar {
187     margin-top: 40px;
188 }
189
190 .banner img {
191     border-radius: 15px;
192 }
193
194 .contributor,
195 .category {
196     padding: 30px 20px;
197     background: #fff;
198     border-radius: 15px;
199     box-shadow: 0 3px 6px #f5f5f5;
200     margin-top: 30px;
201 }
202
203 .contributor__icon {
204     width: 150px;
205     height: 150px;
206     background: #b31566;
207     border-radius: 50%;
208     display: flex;
209     justify-content: center;
210     align-items: center;
211     margin: 0 auto;
212 }
213
214 .contributor__name {
215     text-align: center;
216     margin: 20px 0;
217     font-size: 22px;
218 }

```



```

219
220 .category__ttl {
221     font-size: 22px;
222     margin-bottom: 20px;
223 }
224
225 .category__link {
226     display: block;
227     padding: 10px 0;
228     border-bottom: 1px solid #f5f5f5;
229     position: relative;
230 }
231
232 .category__link::before {
233     position: absolute;
234     top: 15px;
235     right: 10px;
236     content: "";
237     width: 20px;
238     height: 20px;
239     background: url(../img/arrow.svg) no-repeat;
240     background-size: cover;
241 }
242
243 .footer {
244     text-align: center;
245     height: 60px;
246     line-height: 60px;
247     font-size: 14px;
248     background: #fff;
249 }
250

```


cssの設定⑤

- blog_style.cssを作成し、下記のソースを書き写してください。

```

250
251 /* -----categoryページ----- */
252
253 .category-firstview {
254     height: 300px;
255     background-color: #b31566;
256     position: relative;
257 }
258
259 .category__fv-ttl {
260     position: absolute;
261     top: 50%;
262     left: 50%;
263     transform: translate(-50%, -50%);
264     font-size: 60px;
265     color: #fff;
266     font-family: "Indie Flower", cursive;
267 }
268
269 .article__header {
270     padding: 40px 20px;
271     background: #fff;
272     margin-top: 60px;
273 }
274
275 .single__container {
276     background: #fff;
277     border-radius: 15px;
278     max-width: 780px;
279     margin: 0px auto 60px;
280     width: 90%;
281 }
282
283 .article__header {
284     border-radius: 15px 15px 0 0;
285 }
286

```



```

287 .article__eyecatch img {
288     border-radius: 10px;
289 }
290
291 .single__container h1 {
292     margin-top: 30px;
293     font-size: 24px;
294 }
295
296 .single__cat {
297     padding: 5px 20px;
298     background: #b31566;
299     border-radius: 15px;
300 }
301
302 .single__cat a {
303     color: #fff;
304 }
305
306 .tag__ul {
307     margin-top: 30px;
308 }
309
310 .article__content {
311     padding: 40px 15px 60px;
312 }
313
314 .article__content p:first-child {
315     margin-top: 0;
316 }
317
318 .article__content p {
319     margin-top: 20px;
320 }

```


cssの設定⑥

- blog_style.cssを作成し、下記のソースを書き写してください。

```

321
322 .article__content h2 {
323     margin-top: 25px;
324     font-size: 22px;
325 }
326
327 .article__content h3 {
328     margin-top: 30px;
329     font-size: 20px;
330     padding-left: 20px;
331     position: relative;
332 }
333
334 .article__content h3::before {
335     position: absolute;
336     top: 0;
337     left: 0;
338     content: "";
339     width: 10px;
340     height: 100%;
341     background: #b31566;
342 }
343
344 .article__content h4 {
345     margin-top: 25px;
346     font-size: 18px;
347 }
    
```

cssの設定⑦

- pc_style.cssを作成し、下記のソースを書き写してください。

```

pc_style.css x
1 @media screen and (min-width: 760px) {
2   .pc {
3     display: block;
4   }
5
6   .header__nav .flex__item {
7     justify-content: center;
8     margin-top: 20px;
9   }
10
11  .nav__list {
12    color: #594A4E;
13    font-weight: bold;
14    margin-left: 50px;
15  }
16
17  .nav__list:first-child {
18    margin-left: 0;
19  }
20
21  .container {
22    max-width: 1200px;
23  }
24
25  .article-wrap {
26    width: 65%;
27    border-bottom: 0;
28  }
29
30  .sidebar {
31    width: 30%;
32    margin-top: 0;
33  }
34
35  .media-firstview {
36    background: url(../images/background.jpg) no-repeat;
37    background-size: cover;
38    height: 400px;
39  }
40
41  .media__fv-ttl {
42    font-size: 60px;
43  }
44
45  .article__card {
46    width: 30%;
47    margin: 30px 0 0;
48  }
49
50  .article__card:nth-child(-n + 3) {
51    margin-top: 0;
52  }
53
54  /* カテゴリーページ */
55  .category-firstview {
56    height: 200px;
57  }
58
59  /* singleページ */
60
61  .single__container h1 {
62    font-size: 32px;
63  }
64
65  .article__content h2 {
66    font-size: 26px;
67  }
68
69  .article__content h3 {
70    font-size: 22px;
71  }
72
73  .article__content h4 {
74    font-size: 20px;
75  }

```

動作確認①

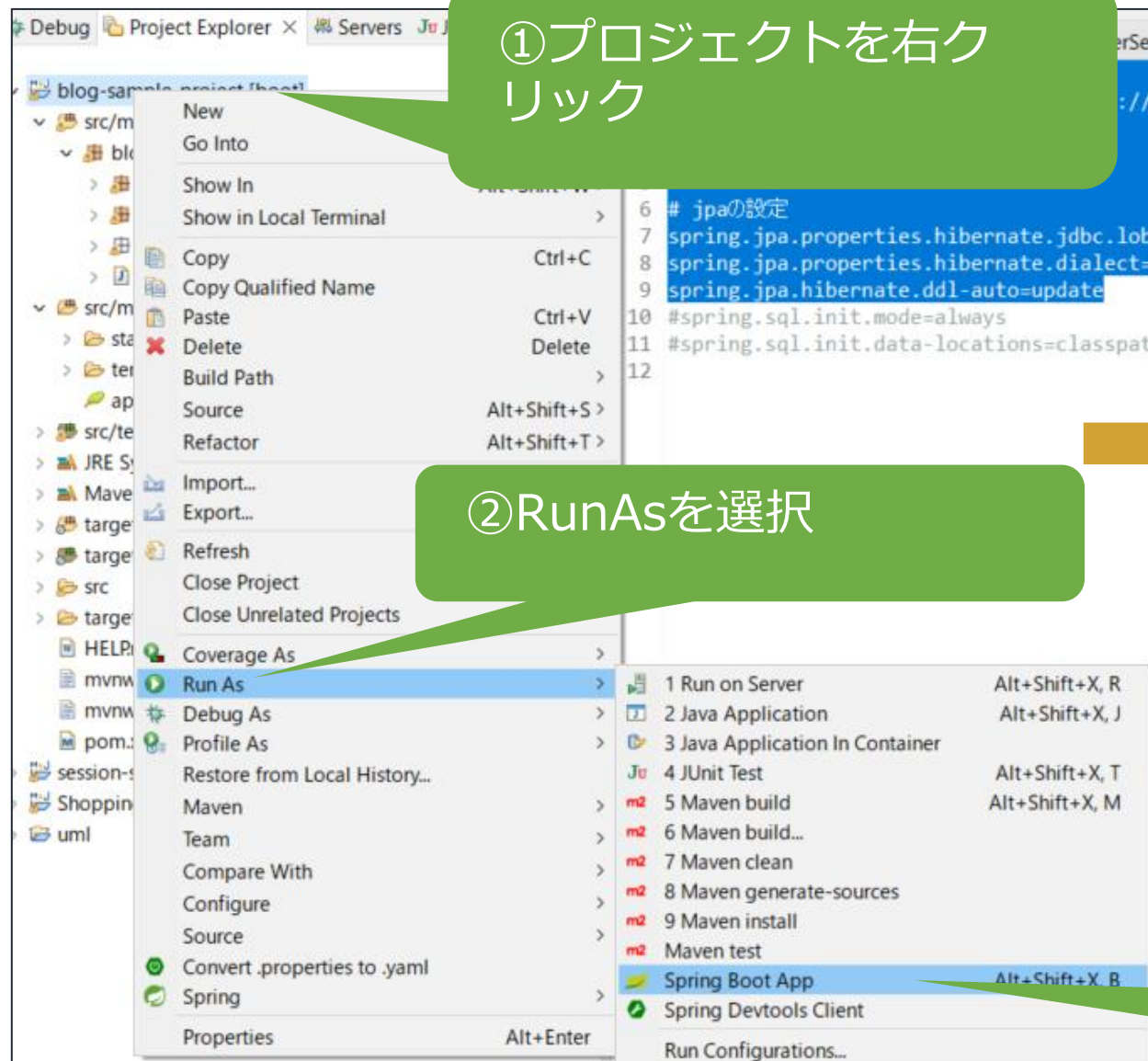
● ユーザー側の動作確認

①プロジェクトを右クリック

②RunAsを選択

③SpringBootApplicationをクリック

④エラーがないかを確認

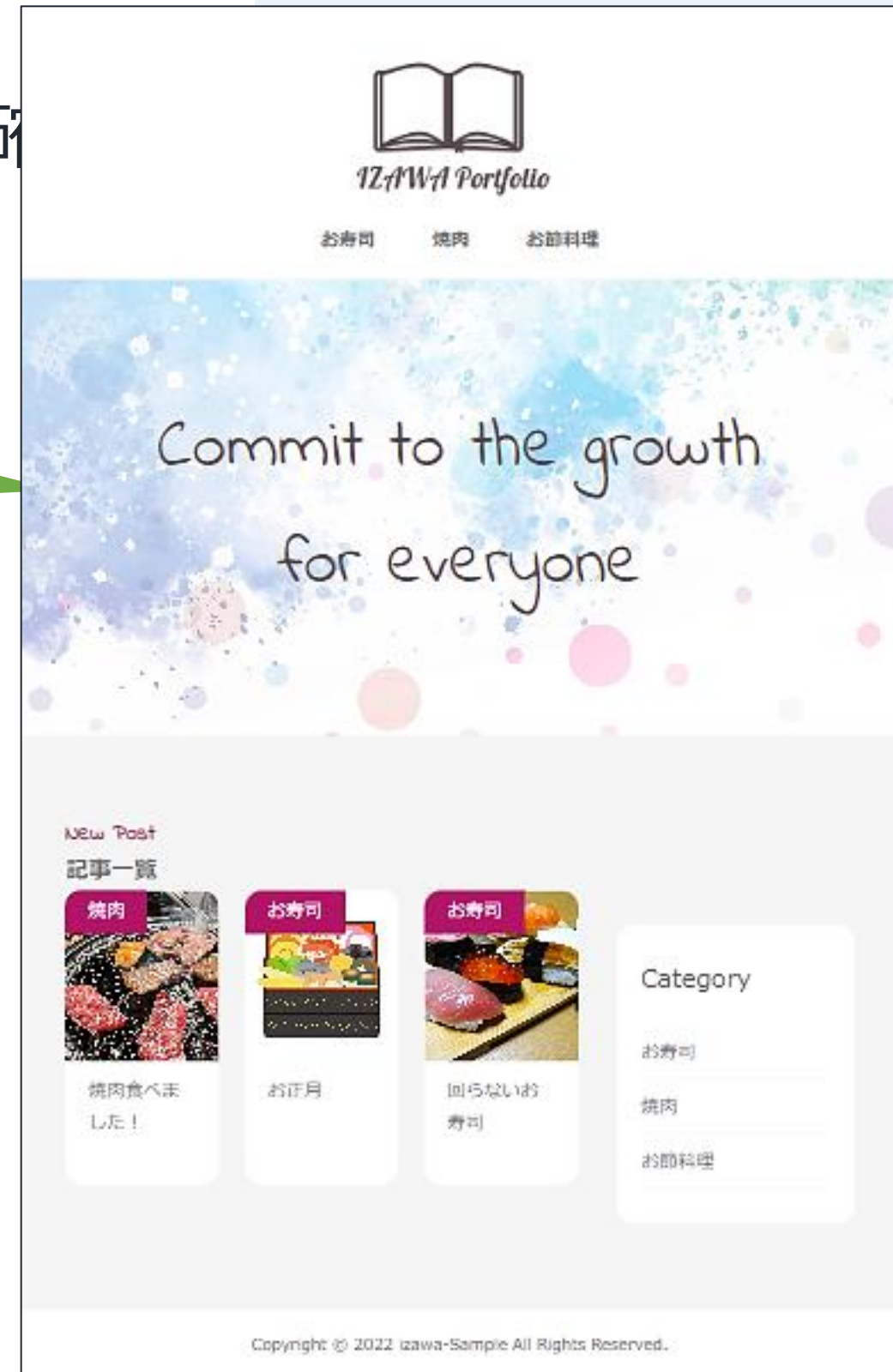


動作確認②

- ブロッグ一覧画面が見れるかを確

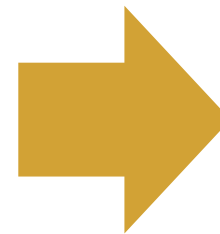
①ブラウザの検索欄に
以下のURLを入力Enter

<http://localhost:8080/blog/>



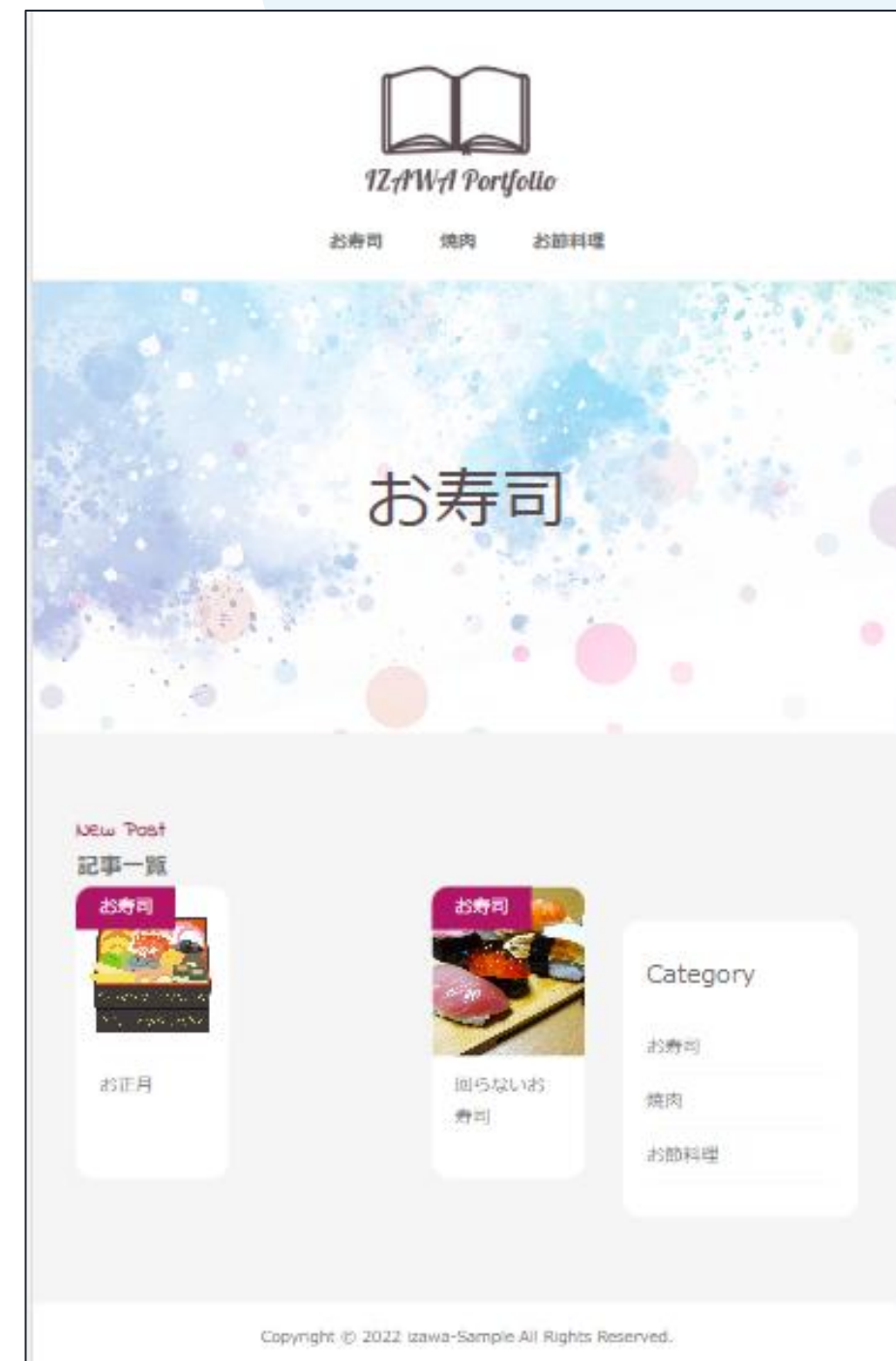
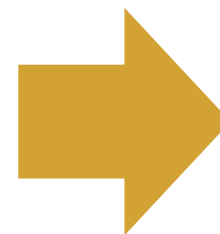
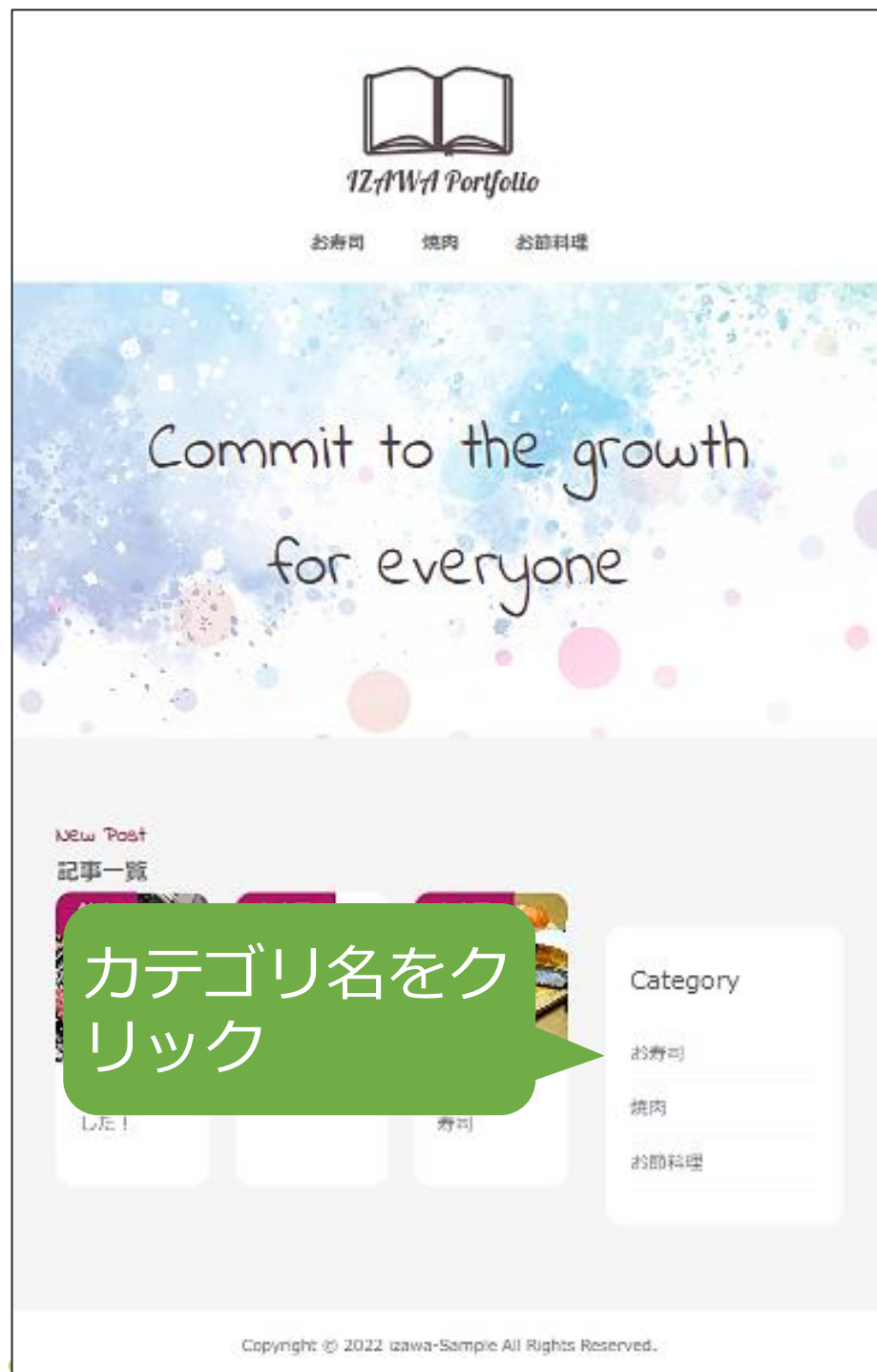
動作確認③

- ブログ詳細画面が見れるかを確認



動作確認④

- カテゴリ別で見れるかを確認



完成

- 以上が個人開発演習のブログの模倣の内容となります。
 - 模倣した内容のgithubURL
 - <https://github.com/Izwa-Akemi/LHP-PersonProject>
- 模倣が終わった人は、ユーザーの使いやすさ（ユーザーインターフェース）を向上させるには、どうすればよいか考え、実践してみてください。（発表会時に質問します）
- ユースケース図や画面遷移図等を自分で考え作成し、発表に備えてください。

Thank you!

From Seeds to Woodland — Shape Your Future.



Shape Your Future