



NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

Practical Report

Web Technology

Computer Science Engineering (Internet of Things)

Semester 3

Kushagra Lakhwani

2021UCI8036

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

November 29, 2022

Contents

1	Using HTML	1
1.1	Project	1
1.2	Source Code	1
2	CSS	2
2.1	Project	2
2.2	Source Code	3
3	Form With Validation	7
3.1	Project	7
3.2	Source Code	7
4	Using HTML, CSS & JavaScript	12
4.1	Project	12
4.2	Source Code	12
5	XML	15
5.1	Project	15
5.2	Source Code	16
6	PHP	17
6.1	Project	17
6.2	Source Code	17

1 Using HTML

1.1 Project

Project Description

This is a simple webpage that showcases the usage of the *HTML* language.

Project Showcase

Welcome

This is just an example with some web content. This is the Hero Unit.

Featured Content 1

lorem dolor ametlorem ipsum dolor ametlorem ipsum dolor ametlorem ipsum.

Featured Content 2

lorem ipsum dolor amet lorem ipsum dolor amet.

©2022 KorigamiK Campos Arias, All Rights Reserved.

Figure 1: Simple Webpage

1.2 Source Code

File: page.html

```
1  <!DOCTYPE HTML>
2  <html lang="en-US">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>My Pimpin Website</title>
7      <meta name="description" content="A sample website, nothin fancy">
8      <meta http-equiv="author" content="Simplistic webpage">
9      <meta name="keywords" content="html, css, web, design, sample,
10         ↪ practice">
11 </head>
12
13 <body>
14     <div class="container">
15         <header>
16             <div class="header">
17                 <h1>Welcome</h1>
```

```

17     </div>
18 </header>
19 <div class="main">
20     <h2>This is just an example with some web content. This is
    ↪ the Hero Unit.</h2>
21 </div>
22 <div class="feature">
23     <h3>Featured Content 1</h3>
24     <p>lorem dolor ametlorem ipsum dolor ametlorem ipsum dolor
    ↪ ametlorem ipsum.</p>
25 </div>
26 <div class="feature">
27     <h3>Featured Content 2</h3>
28     <p>lorem ipsum dolor amet lorem ipsum dolor ametr.</p>
29 </div>
30 <footer>
31     &copy;2022 KorigamiK Campos Arias, All Rigts Reserved.
32 </footer>
33 </div>
34 </body>
35
36 </html>

```

2 CSS

2.1 Project

Project Description

This is a simple web-page that utilizes an external cascade stylesheet

Project Showcase

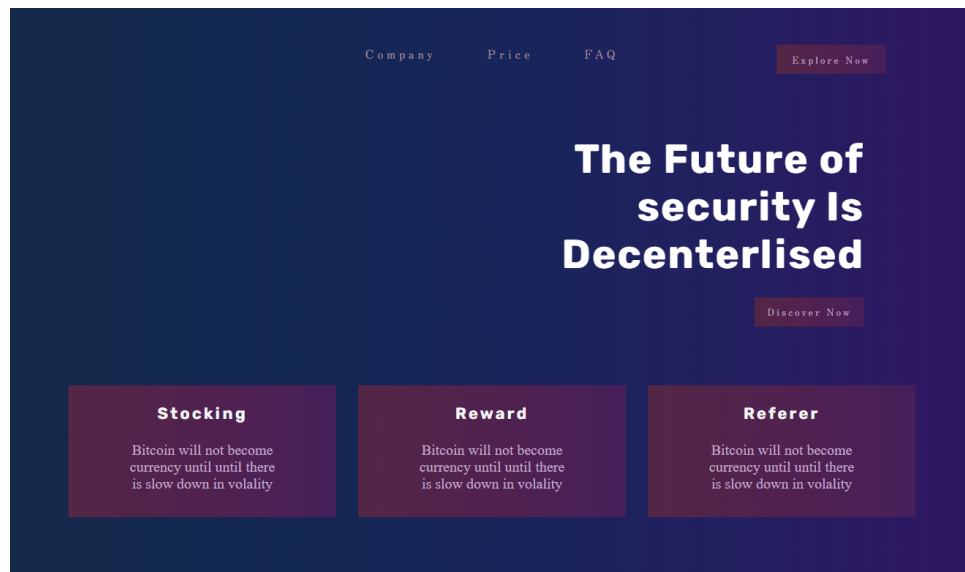


Figure 2: Styled Web page

2.2 Source Code

File: page.css

```
1  @import
   ↳ url('https://fonts.googleapis.com/css2?family=Zen+Old+Mincho&display=swap');
2  @import
   ↳ url('https://fonts.googleapis.com/css2?family=Zen+Old+Mincho:wght@900&display=sw
3  @import
   ↳ url('https://fonts.googleapis.com/css2?family=Rubik:wght@700&display=swap');
4
5  body {
6      background-image: linear-gradient(to right, #152a49, #132852,
   ↳ #18255a,
7          #232060, #311763);
8  }
9
10 .nav {
11     margin: 50px;
12     list-style: none;
13     display: flex;
14     justify-content: center;
15     gap: 20px;
16     color: #A8999C;
17     letter-spacing: 5px;
```

```

18 }
19
20 .cnav {
21     margin: 0px 25px;
22     font-family: 'Zen Old Mincho', serif;
23 }
24
25 .b1 {
26     background-image: linear-gradient(to right, #552746, #53244a,
27     ↪ #51224f,
28         #4c2155, #46205b);
29     height: 40px;
30     width: 150px;
31     color: #C3AFCC;
32     border: none;
33     font-family: 'Zen Old Mincho', serif;
34     position: absolute;
35     top: 50px;
36     right: 120px;
37     letter-spacing: 3px;
38     font-weight: 100;
39 }
40
41 .content {
42     color: white;
43     font-family: 'Rubik', sans-serif;
44     font-size: 55px;
45     text-align: end;
46     letter-spacing: 2px;
47     padding-left: 800px;
48     margin-right: 150px;
49     margin-top: 100px;
50 }
51
52 .b2 {
53     background-image: linear-gradient(to right, #552746, #53244a,
54     ↪ #51224f,
55         #4c2155, #46205b);
56     height: 40px;
57     width: 150px;
58     color: #C3AFCC;
59     border: none;
60     letter-spacing: 3px;
61     font-family: 'Zen Old Mincho', serif;
62 }

```

```

61
62 .cb {
63     display: flex;
64     gap: 30px;
65     justify-content: center;
66     margin: 80px;
67 }
68
69 .cb1 {
70     background-image: linear-gradient(to right, #552746, #53244a,
71     ↪ #51224f,
72     #4c2155, #46205b);
73     height: 180px;
74 }
75
76 .h {
77     text-align: center;
78     font-family: 'Rubik', sans-serif;
79     margin: 25px;
80     font-size: 22px;
81     color: whitesmoke;
82     letter-spacing: 3px;
83 }
84
85 .pc {
86     text-align: center;
87     padding: 0px 75px;
88     font-size: 20px;
89     color: #C3AFCC;
90     letter-spacing: 2px;
91 }

```

File: page.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8" />
6      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7      <meta name="viewport" content="width=device-width,
8      ↪ initial-scale=1.0" />
9      <title>Nft Marketplace</title>

```

```

9
10 <head>
11     <link rel="stylesheet" href="page.css" />
12 </head>
13
14 <body>
15     <nav>
16         <ul class="nav">
17             <li class="cnav">Company</li>
18             <li class="cnav">Price</li>
19             <li class="cnav">FAQ</li> <button class="b1">Explore
                ↪ Now</button>
20         </ul>
21     </nav>
22     <div class="content">The Future of security Is Decenterlised <br>
        ↪ <button class="b2">Discover Now</button></div>
23     <div class="cb">
24         <div class="cb1">
25             <p class="h">Stocking</p>
26             <p class="pc">
27                 Bitcoin will not become currency until until there is
                ↪ slow
28                 down in volality
29             </p>
30         </div>
31         <div class="cb1">
32             <p class="h">Reward</p>
33             <p class="pc">
34                 Bitcoin will not become currency until until there is
                ↪ slow
35                 down in volality
36             </p>
37         </div>
38         <div class="cb1">
39             <p class="h">Referer</p>
40             <p class="pc">
41                 Bitcoin will not become currency until until there is
                ↪ slow
42                 down in volality
43             </p>
44         </div>
45     </div>
46 </body>

```



```
47  
48 </html>
```

3 Form With Validation

3.1 Project

Project Description

This project is a *Signup form with validation*.

Project Showcase

Sign Up

Name

asd

Email

as@qw

Password

...

Submit

Password must contain the following:

- A lowercase letter
- A capital (uppercase) letter
- A number
- Minimum 8 characters

© 2022 - 2023 KorigamiK

Figure 3: Form With Validation

3.2 Source Code

File: form.html

```

1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6      <meta name="viewport" content="width=device-width,
7        ↪ initial-scale=1.0" />
8      <title>SignUp Form</title>
9      <style>
10         :root {
11           --message-display: none;
12         }
13         body {
14           font-family: "Courier New", Courier, monospace;
15         }
16         main {
17           width: 100%;
18           height: 100vh;
19           display: flex;
20           flex-direction: column;
21           justify-content: center;
22           align-items: center;
23         }
24         form {
25           text-align: center;
26         }
27         form .user-input {
28           display: grid;
29           grid-template-columns: 1fr 1fr;
30           align-items: center;
31           border: 1px solid black;
32           border-radius: 10px;
33           padding: 20px;
34         }
35         input {
36           height: 30px;
37           margin: 10px 0;
38           border-radius: 5px;
39           padding: 5px;
40         }
41         button {
42           margin: 10px 0;
43           border: 1px solid black;
44           border-radius: 5px;

```

```

44     padding: 5px 10px;
45     background-color: #f1f1f1;
46 }
47 button:hover {
48     background-color: #ddd;
49 }
50 input:invalid {
51     border: 2px solid red;
52 }
53 input:valid {
54     border: 2px solid green;
55 }
56 #message {
57     display: var(--message-display);
58 }
59 .valid {
60     color: green;
61 }
62 .invalid {
63     color: red;
64 }
65 footer {
66     width: 100%;
67     text-align: center;
68     text-decoration: none;
69 }
70 footer a {
71     text-decoration: none;
72 }
73 </style>
74 </head>
75 <body>
76     <main>
77         <h1>Sign Up</h1>
78         <form action="/signup" method="POST">
79             <div class="user-input">
80                 <label for="name">Name</label>
81                 <input required type="text" name="name" id="name" />
82                 <label for="email">Email</label>
83                 <input required type="email" name="email" id="email" />
84                 <label for="password">Password</label>
85                 <input
86                     required
87                     type="password"
88                     name="password"

```

```

89         pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}"
90         id="password"
91     />
92 </div>
93 <button type="submit">Submit</button>
94 </form>
95
96 <div id="message">
97     <h3>Password must contain the following:</h3>
98     <p id="letter" class="invalid">A <b>lowercase</b> letter</p>
99     <p id="capital" class="invalid">A <b>capital (uppercase)</b>
100       ↳ letter</p>
101     <p id="number" class="invalid">A <b>number</b></p>
102     <p id="length" class="invalid">Minimum <b>8 characters</b></p>
103 </div>
104 </main>
105 <footer>
106     &copy; 2022 - 2023
107     <a target="_blank"
108       ↳ href="https://github.com/KorigamiK">KorigamiK</a>
109 </footer>
110
111 <script type="text/JavaScript">
112     const form = document.querySelector("form");
113     form.addEventListener("submit", (e) => {
114         e.preventDefault();
115     });
116
117     const password = document.getElementById("password");
118     password.addEventListener("blur", () =>
119       ↳ document.documentElement.style.setProperty("--message-display",
120       ↳ "none"));
121     password.addEventListener("focus", () =>
122       ↳ document.documentElement.style.setProperty("--message-display",
123       ↳ "block"));
124
125     const message = document.getElementById("message");
126     const letter = document.getElementById("letter");
127     const capital = document.getElementById("capital");
128     const number = document.getElementById("number");
129     const length = document.getElementById("length");
130
131     password.onkeyup = () => {
132         const lowerCaseLetters = /[a-z]/g;
133         if (password.value.match(lowerCaseLetters)) {

```

```

128         letter.classList.toggle("invalid");
129         letter.classList.add("valid");
130     } else {
131         letter.classList.remove("valid");
132         letter.classList.add("invalid");
133     }
134
135     const upperCaseLetters = /[A-Z]/g;
136     if (password.value.match(upperCaseLetters)) {
137         capital.classList.remove("invalid");
138         capital.classList.add("valid");
139     } else {
140         capital.classList.remove("valid");
141         capital.classList.add("invalid");
142     }
143
144     const numbers = /[0-9]/g;
145     if (password.value.match(numbers)) {
146         number.classList.remove("invalid");
147         number.classList.add("valid");
148     } else {
149         number.classList.remove("valid");
150         number.classList.add("invalid");
151     }
152
153     if (password.value.length >= 8) {
154         length.classList.remove("invalid");
155         length.classList.add("valid");
156     } else {
157         length.classList.remove("valid");
158         length.classList.add("invalid");
159     }
160     };
161 </script>
162 </body>
163 </html>

```

4 Using HTML, CSS & JavaScript

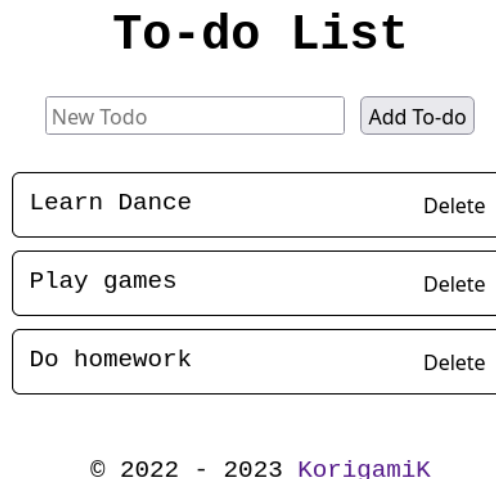
4.1 Project

Project Description

This is a minimal To-do List application written in HTML, CSS and JavaScript. It is a single page application that allows the user to add, edit and delete tasks.

The tasks are added through a form that is displayed at the top of the page. The tasks are displayed in an unordered list below the form. And a delete button is displayed next to each task. The click and submit events are handled by JavaScript functions.

Project Showcase



The screenshot shows a web application titled "To-do List". At the top, there is a text input field labeled "New Todo" and a button labeled "Add To-do". Below this, there is a list of three tasks, each in a rounded rectangular box. The first task is "Learn Dance" with a "Delete" button to its right. The second task is "Play games" with a "Delete" button to its right. The third task is "Do homework" with a "Delete" button to its right. At the bottom of the page, there is a copyright notice: "© 2022 - 2023 KorigamiK".

Figure 4: To-do List App

4.2 Source Code

File: todo-list.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
```

```

4 <meta charset="UTF-8" />
5 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6 <meta name="viewport" content="width=device-width,
  ↳ initial-scale=1.0" />
7 <style>
8   body {
9     font-family: "Courier New", Courier, monospace;
10  }
11  footer {
12    position: static;
13    width: 100%;
14    text-align: center;
15    text-decoration: none;
16    margin-top: 1rem;
17  }
18  footer a {
19    text-decoration: none;
20  }
21  .todo-list {
22    display: flex;
23    flex-direction: column;
24    align-items: center;
25    justify-content: space-between;
26    margin-top: 1rem;
27  }
28  .todo-list ul {
29    list-style: none;
30    padding: 0;
31  }
32  .todo-list li {
33    display: flex;
34    flex-direction: row;
35    justify-content: space-between;
36    max-width: 500px;
37    min-width: 300px;
38    margin: 0.5rem;
39    padding: 10px;
40    border: 1px solid black;
41    border-radius: 5px;
42  }
43  .todo-list li button {
44    border: none;
45    background-color: transparent;
46    cursor: pointer;
47    margin-left: 0.5rem;

```

```

48     }
49     .todo-list li button:hover {
50         color: red;
51     }
52 </style>
53 <title>To-do List</title>
54 </head>
55 <body>
56     <main class="todo-list">
57         <h1>To-do List</h1>
58
59         <form class="new-todo__form" action="/new-todo" method="POST">
60             <input type="text" name="todo" placeholder="New Todo" />
61             <button type="submit">Add To-do</button>
62         </form>
63
64         <ul>
65             <li>
66                 <span>Learn Node.js</span>
67                 <button>Delete</button>
68             </li>
69         </ul>
70     </main>
71
72     <footer>
73         &copy; 2022 - 2023
74         <a target="_blank"
75             ↪ href="https://github.com/KorigamiK">KorigamiK</a>
76     </footer>
77
78     <script type="text/javascript">
79         const deleteButtons = document.querySelectorAll(".todo-list li
80             ↪ button");
81
82         deleteButtons.forEach((button) => {
83             button.addEventListener("click", (event) => {
84                 event.target.parentElement.remove();
85             });
86         });
87
88         const todoList = document.querySelector(".todo-list > ul");
89
90         const addTodo = (todo) => {
91             const todoItem = document.createElement("li");
92             const todoText = document.createElement("span");

```



```

91     todoText.innerText = todo;
92
93     const deleteButton = document.createElement("button");
94     deleteButton.innerText = "Delete";
95
96     deleteButton.addEventListener("click", (e) =>
97         e.target.parentElement.remove()
98     );
99
100    todoItem.appendChild(todoText);
101    todoItem.appendChild(deleteButton);
102
103    todoList.appendChild(todoItem);
104 };
105
106 const newTodoForm = document.querySelector(".new-todo__form");
107
108 newTodoForm.addEventListener("submit", (event) => {
109     event.preventDefault();
110
111     const formData = new FormData(event.target);
112     const todo = formData.get("todo");
113     todo.length ? addTodo(todo) : alert("Please enter a todo");
114
115     event.target.reset();
116 });
117
118 addTodo("Learn React");
119 </script>
120 </body>
121 </html>

```

5 XML

5.1 Project

Project Description

This is a syndication feed for my blog written in *XML* following the Atom standard for syndication.

5.2 Source Code

File: feed.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <feed xmlns="http://www.w3.org/2005/Atom">
3      <id>https://korigamik.deno.dev/blog</id>
4      <title>Blogs on paper</title>
5      <updated>2022-11-02T06:46:07.509Z</updated>
6      <link rel="alternate" href="https://korigamik.deno.dev/blog"/>
7      <link rel="self" href="https://korigamik.deno.dev/feed"/>
8      <subtitle>Articles thrown at you.</subtitle>
9      <icon>https://korigamik.deno.dev/favicon.ico</icon>
10     <rights>Copyright 2022 https://korigamik.deno.dev</rights>
11     <entry>
12         <title type="html"><![CDATA[Bags with Balls]]></title>
13         <id>https://korigamik.deno.dev/Bags with Balls</id>
14         <link href="https://korigamik.deno.dev/bags_with_balls"/>
15         <updated>2022-08-16T13:15:18.514Z</updated>
16         <summary type="html"><![CDATA[An approach to solving this
17             ↪ codeforces problem]]></summary>
18         <published>2022-08-16T13:15:18.514Z</published>
19         <rights>Copyright 2022 https://korigamik.deno.dev</rights>
20     </entry>
21     <entry>
22         <title type="html"><![CDATA[Hello world!]]></title>
23         <id>https://korigamik.deno.dev/Hello world!</id>
24         <link href="https://korigamik.deno.dev/hello_world"/>
25         <updated>2022-08-15T19:15:18.514Z</updated>
26         <summary type="html"><![CDATA[This is my first blog post!
27             ↪ ]]]></summary>
28         <published>2022-08-15T19:15:18.514Z</published>
29         <rights>Copyright 2022 https://korigamik.deno.dev</rights>
30     </entry>
31     <entry>
32         <title type="html"><![CDATA[Raycasting Adventures]]></title>
33         <id>https://korigamik.deno.dev/Raycasting Adventures</id>
34         <link
35             ↪ href="https://korigamik.deno.dev/raycasting_adventures"/>
36         <updated>2022-10-25T13:15:18.514Z</updated>
37         <summary type="html"><![CDATA[A deep dive into
38             ↪ game...]]></summary>
39         <published>2022-10-25T13:15:18.514Z</published>
40         <rights>Copyright 2022 https://korigamik.deno.dev</rights>
```

```
37     </entry>
38 </feed>
```

6 PHP

6.1 Project

Project Description

This is a simple *PHP* application[1, GitHub] that displays a list of fruits. The fruits are stored in a SQLite database.

The user can add more fruits to the list by filling a form and submitting it and the user can also buy the fruits by clicking on the *Buy* button.

The fruit is removed from the list when none of its kind are left.

Project Showcase

Shop Fruits

Name:

Amount:

Fruits

Name	Amount	Buy
Apple	3	Buy
Banana	2	Buy

Figure 5: PHP Fruits App

6.2 Source Code

File: index.php

```

1 <!DOCTYPE html>
2
3 <?php include 'app.php'; ?>
4
5 <html>
6 <head>
7     <title>Fruit Shop</title>
8 </head>
9 <body>
10     <h1> Shop Fruits </h1>
11     <form action="/api/add.php" method="post">
12         <label for="name">Name:</label>
13         <input type="text" id="name" name="name"><br><br>
14         <label for="amount">Amount:</label>
15         <input type="text" id="amount" name="amount"><br><br>
16         <input type="submit" value="Submit">
17     </form>
18
19     <h2> Fruits </h2>
20     <table>
21         <tr>
22             <th>Name</th>
23             <th>Amount</th>
24             <th>Buy</th>
25         </tr>
26         <?php
27         $results = $db->GetFruits();
28         while ($row = $results->fetchArray()) {
29             echo "<tr>";
30             echo "<td>" . $row['name'] . "</td>";
31             echo "<td>" . $row['amount'] . "</td>";
32             echo "<td><a href='/api/buy.php?id=" . $row['id'] .
33                 ↵ "'>Buy</a></td>";
34             echo "</tr>";
35         }
36     </table>
37 </body>
38 </html>

```

File: app.php

```

1  <?php
2  class MyDB extends SQLite3
3  {
4      public $rootDir;
5      function __construct()
6      {
7          $this->rootDir = realpath($_SERVER["DOCUMENT_ROOT"]);
8          $this->open($this->rootDir . '/test.db');
9      }
10
11     public function CreateTables()
12     {
13         $this->exec('CREATE TABLE IF NOT EXISTS fruits (id INTEGER
14             ↳ PRIMARY KEY, name TEXT, amount INTEGER, CHECK (amount >
15             ↳ 0))');
16     }
17
18     public function AddFruit($name, $amount)
19     {
20         $this->exec("INSERT INTO fruits (name, amount) VALUES
21             ↳ ('$name', '$amount')");
22     }
23
24     public function GetFruits()
25     {
26         $results = $this->query('SELECT * FROM fruits');
27         return $results;
28     }
29
30     public function BuyFruit($id)
31     {
32         $this->exec("UPDATE fruits SET amount = amount - 1 WHERE id =
33             ↳ $id");
34         if ($this->changes() == 0) {
35             $this->exec("DELETE FROM fruits WHERE id = $id");
36         }
37     }
38
39     function __destruct()
40     {
41         $this->close();
42     }
43 }

```

```

41 class Logger
42 {
43     private $logFile;
44
45     function __construct()
46     {
47         $this->logFile = fopen(realpath($_SERVER["DOCUMENT_ROOT"]) .
48             ↪ "/log.txt", "a");
49         $this->Log("Logger created");
50     }
51
52     public function Log($message)
53     {
54         fwrite($this->logFile, $message . PHP_EOL);
55     }
56
57     public function __destruct()
58     {
59         $this->Log("Logger destroyed");
60         fclose($this->logFile);
61     }
62 }
63
64 $logger = new Logger();
65 $db = new MyDB();
66
67 if (!$db) {
68     echo $db->lastErrorMsg();
69 } else {
70     $logger->Log("Opened database successfully");
71     $db->CreateTables();
72 }

```

File: api/add.php

```
1  <?php include '../app.php'; ?>
2
3  <?php
4  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5      $logger->Log("Adding fruit");
6      $db->AddFruit($_POST['name'], $_POST['amount']);
7      header("Location: /");
8      exit();
9  } else {
10     return false;
11 }
12 ?>
```

File: api/buy.php

```
1  <?php include '../app.php'; ?>
2
3  <?php
4  if ($_SERVER['REQUEST_METHOD'] === 'GET') {
5      $logger->Log("Buying fruit with id: " . $_GET['id']);
6      $db->BuyFruit($_GET['id']);
7      exit(header("Location: /"));
8  } else {
9      return false;
10 }
11 ?>
```

References

[1] KorigamiK, “php-fruits.” <https://github.com/korigamiK/php-fruits/>, 2022.

List of Figures

1	Simple Webpage	1
2	Styled Web page	3
3	Form With Validation	7
4	To-do List App	12
5	PHP Fruits App	17