HANDS ON EXERCISE -WEEK 4

**1: Product Management API**

Manage products (Add, Get, Update, Delete) using REST API.

**Product.java**

public class Product {

private int id;

private String name;

private double price;

// Constructors, Getters, Setters

}

**ProductService.java**

@Service

public class ProductService {

private final Map<Integer, Product> productMap = new HashMap<>();

public List<Product> getAll() {

return new ArrayList<>(productMap.values());

}

public Product getById(int id) {

return productMap.get(id);

}

public void add(Product p) {

productMap.put(p.getId(), p);

}

public void update(Product p) {

productMap.put(p.getId(), p);

}

public void delete(int id) {

productMap.remove(id);

}

}

**ProductController.java**

@RestController

@RequestMapping("/products")

public class ProductController {

@Autowired

private ProductService service;

@GetMapping

public List<Product> getAll() {

return service.getAll();

}

@GetMapping("/{id}")

public Product getById(@PathVariable int id) {

return service.getById(id);

}

@PostMapping

public String add(@RequestBody Product p) {

service.add(p);

return "Product added.";

}

@PutMapping

public String update(@RequestBody Product p) {

service.update(p);

return "Product updated.";

}

@DeleteMapping("/{id}")

public String delete(@PathVariable int id) {

service.delete(id);

return "Product deleted.";

}

}

**API Endpoints**

| **METHOD** | **URL** | **FUNCTION** |
| --- | --- | --- |
| GET | /products | Get all products |
| GET | /products/{id} | Get product by ID |
| POST | /products | Add product |
| PUT | /products | Update product |
| DELETE | /products/{id} | Delete product |

**Sample JSON Input**

{

"id": 1,

"name": "Laptop",

"price": 59999.99

}

**2: User Registration API**

Register and view users using a simple REST service.

**User.java**

public class User {

private String username;

private String email;

// Constructors, Getters, Setters

}

**UserController.java**

@RestController

@RequestMapping("/users")

public class UserController {

private final List<User> userList = new ArrayList<>();

@PostMapping("/register")

public String register(@RequestBody User user) {

userList.add(user);

return "User registered!";

}

@GetMapping

public List<User> getAllUsers() {

return userList;

}

}

**API Endpoints**

| **METHOD** | **URL** | **FUNCTION** |
| --- | --- | --- |
| POST | /users/register | Register new user |
| GET | /users | View all users |

**JSON Input Example**

{

"username": "balu",

"email": "balu@example.com"

}

**3: Feedback Submission API**

Submit and retrieve feedback entries.

**Feedback.java**

public class Feedback {

private int id;

private String message;

// Constructors, Getters, Setters

}

**FeedbackController.java**

@RestController

@RequestMapping("/feedback")

public class FeedbackController {

private final List<Feedback> feedbackList = new ArrayList<>();

@PostMapping

public String submit(@RequestBody Feedback feedback) {

feedbackList.add(feedback);

return "Feedback submitted.";

}

@GetMapping

public List<Feedback> viewAll() {

return feedbackList;

}

}

**API Endpoints**

| **METHOD** | **URL** | **FUNCTION** |
| --- | --- | --- |
| POST | /feedback | Submit feedback |
| GET | /feedback | View all feedback |

**Sample JSON Input**

{

"id": 101,

"message": "Great service!"

}