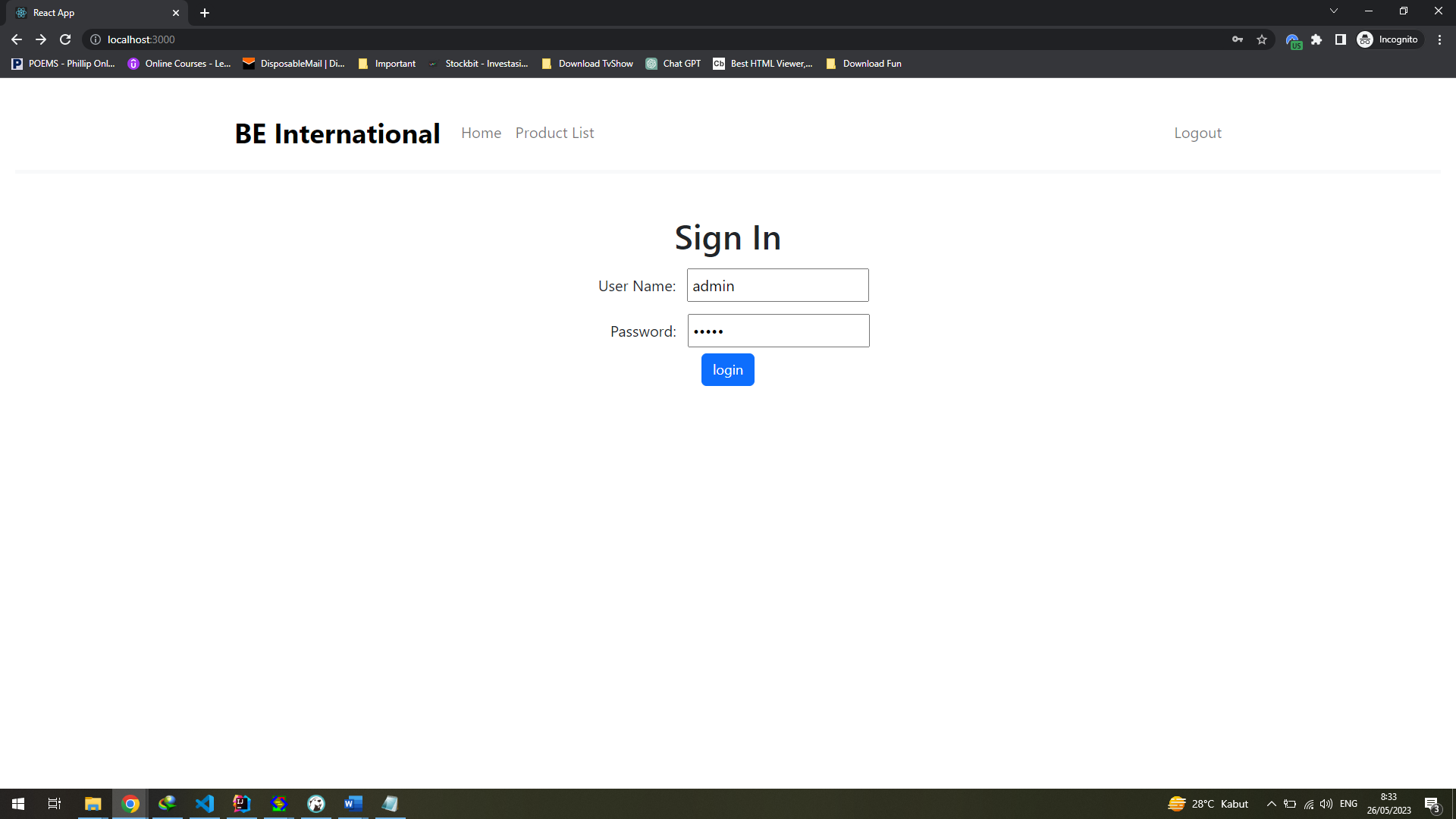
USER GUIDE

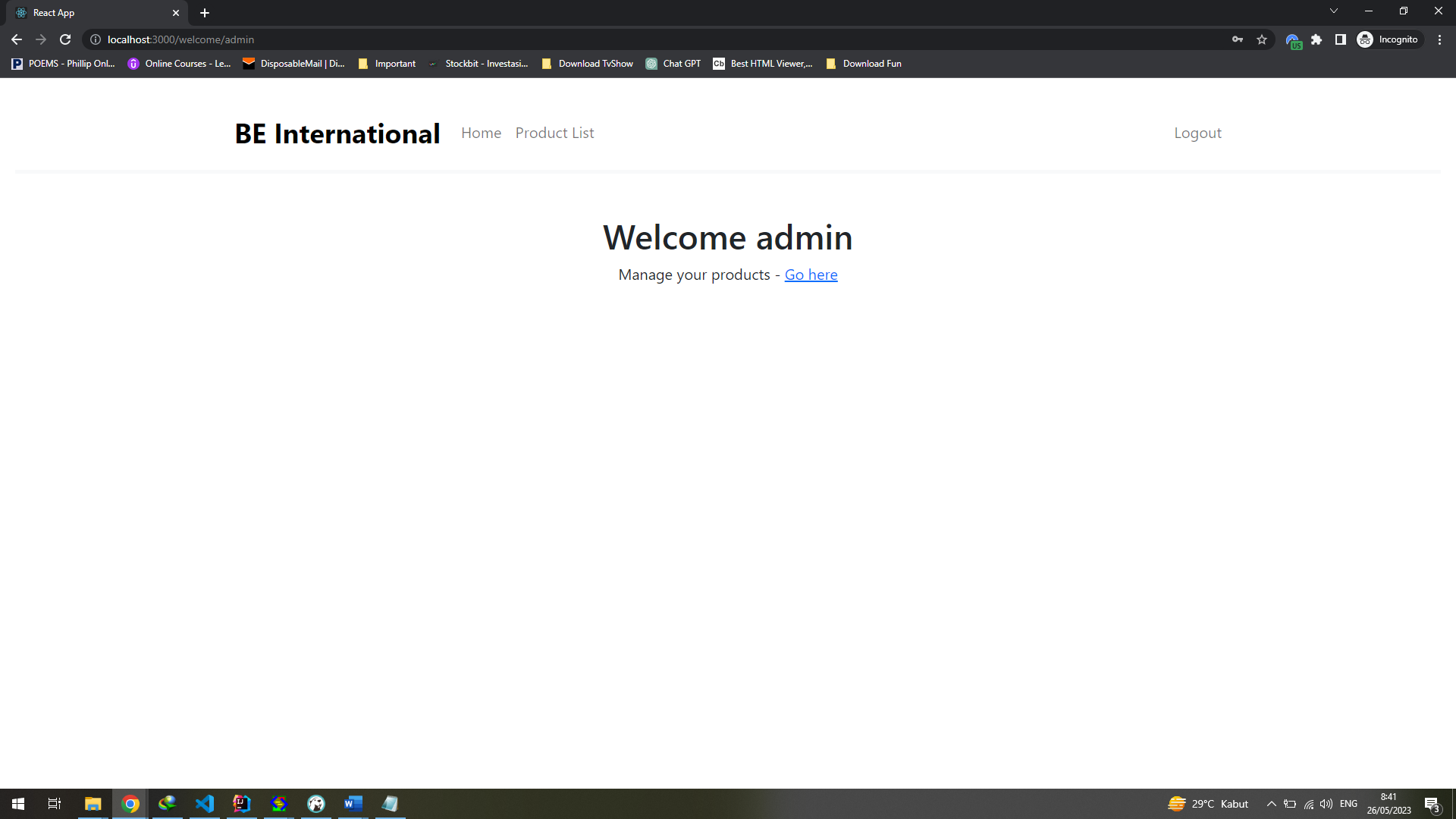
This product app consists of two applications: a front-end app developed using React JS and a back-end app built with Java using the Spring framework. It utilizes MySQL as the database. In case MySQL is not installed or if you prefer not to install it, Docker can be used as an alternative.

Here are the screenshots and guide to run the front-end app:

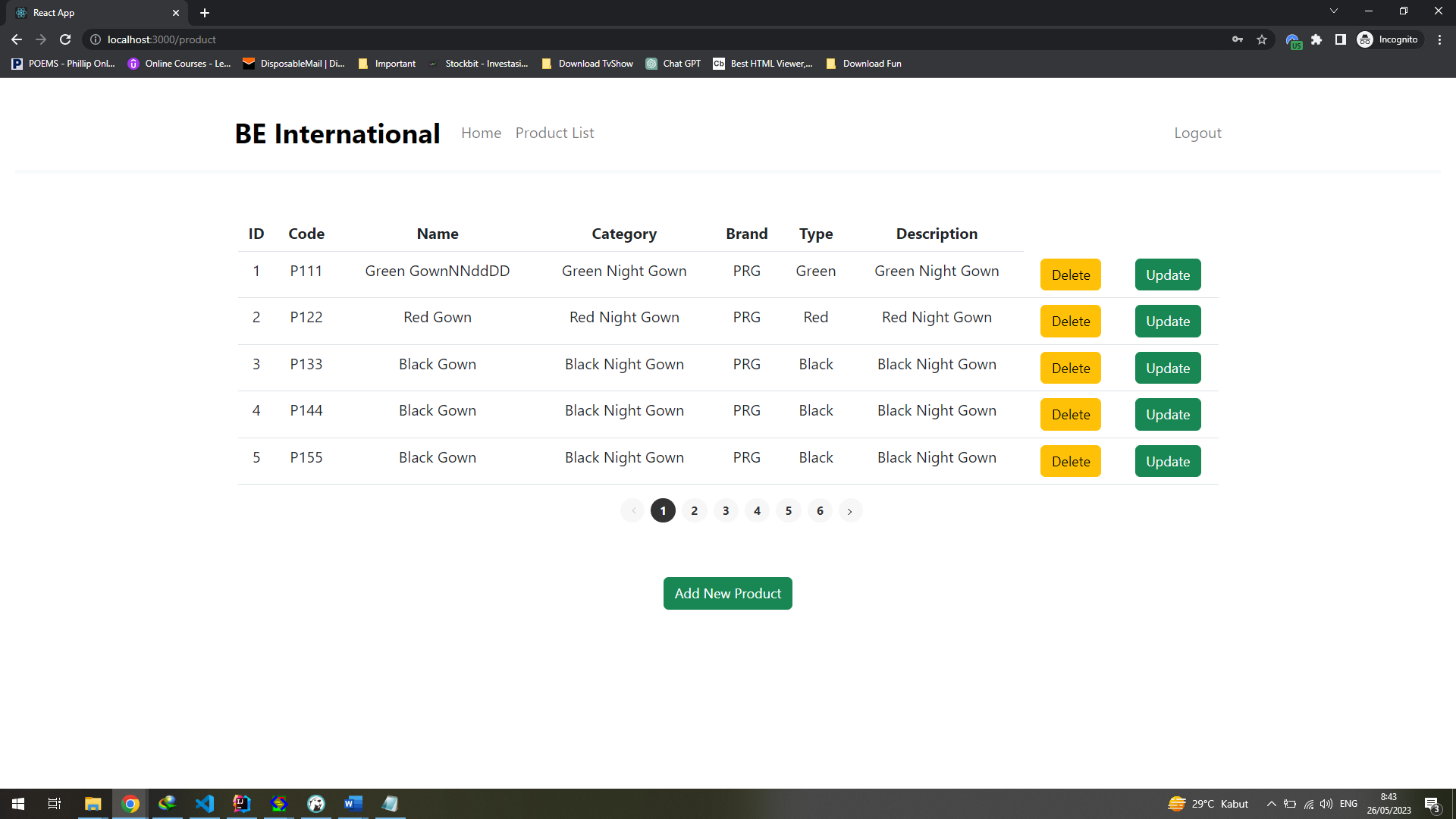
I didn’t upload the ‘node\_modules’ for the React dependencies, so make sure to run ‘npm install’ and then run ‘npm start’ to start the app. The default port is 3000. Here is the welcome page, a simple user login page. The username is "admin," and the password is also "admin". I have hardcoded the username and password to make the login process simpler



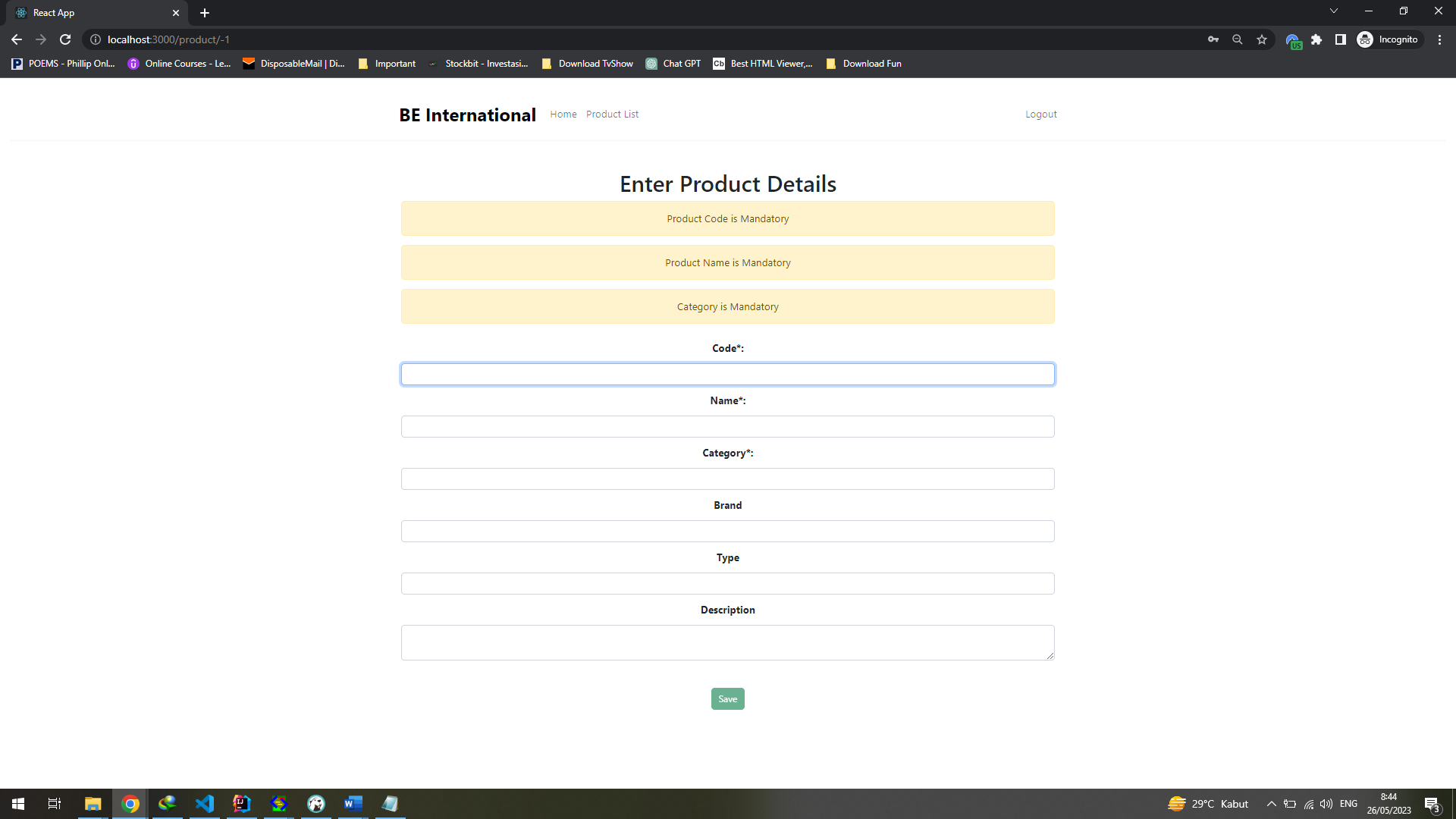
Here is a simple welcome page. Click “Go Here” or “Product List” to see the list of products,

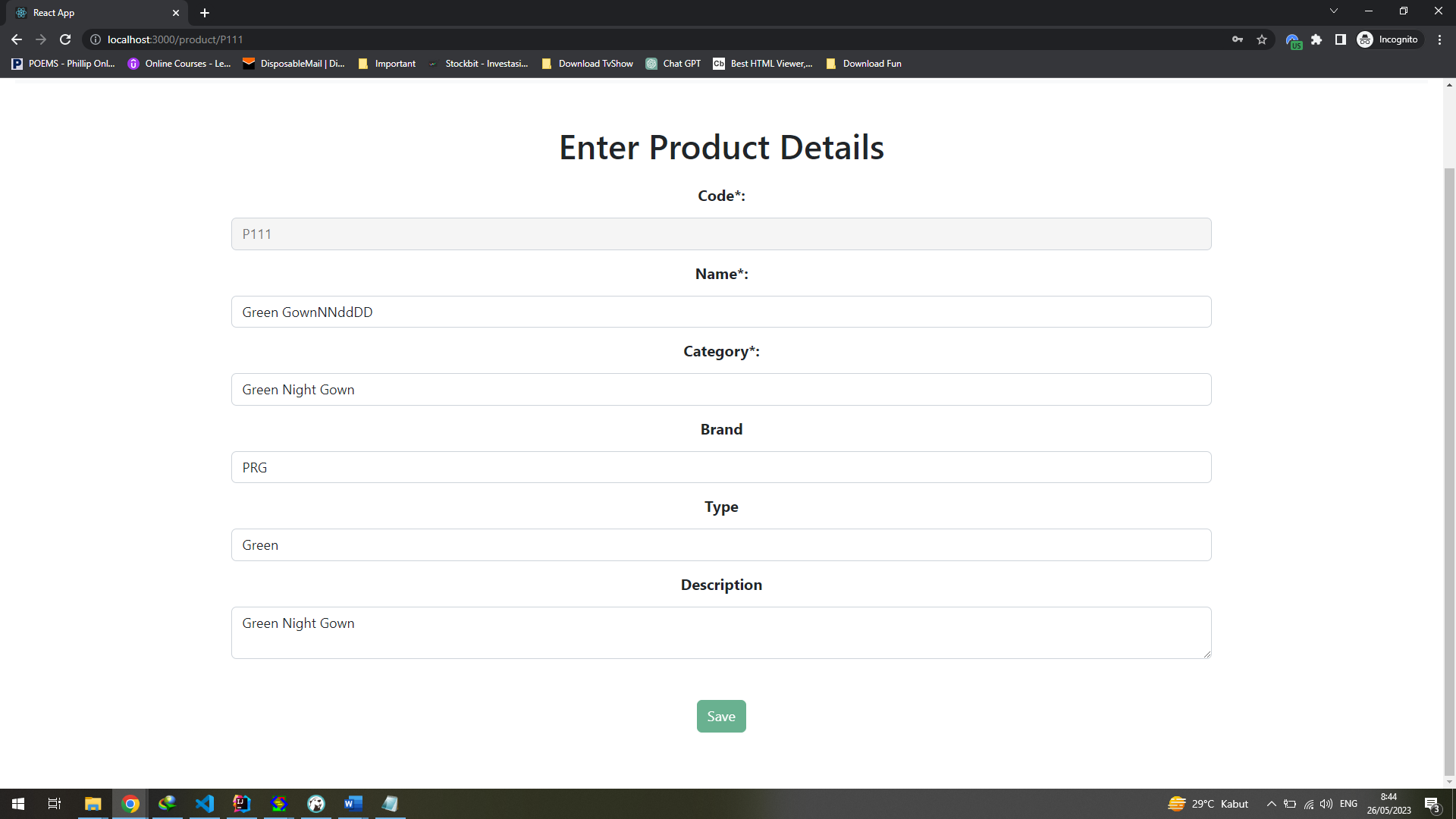


Here is the Product List Page: I have set the page limit to 5 products per page, resulting in multiple pages for pagination.



Here is the Add New Product Page and the Update Page :





Here are the guides for the backend Java app. The app is using Maven, and the database name is "productapp" (refer to the application.properties file for details). The project runs on the default port 8080.

Here, I have included the SQL script to create the table and 25 sample data. Feel free to add more as needed.

USE `productapp`;

CREATE TABLE `product` (

`id` int NOT NULL AUTO\_INCREMENT,

`productcode` varchar(255) DEFAULT NULL,

`productname` varchar(255) DEFAULT NULL,

`category` varchar(100) DEFAULT NULL,

`brand` varchar(100) DEFAULT NULL,

`type` varchar(100) DEFAULT NULL,

`productdesc` varchar(100) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=latin1;

INSERT INTO `product` VALUES

(1,'P111','Green Gown','Green Night Gown','PRG','Green','Green Night Gown'),

(2,'P122','Red Gown','Red Night Gown','PRG','Red','Red Night Gown'),

(3,'P133','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(4,'P144','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(5,'P155','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(6,'P166','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(7,'P177','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(8,'P188','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(9,'P199','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(10,'P211','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(11,'P221','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(12,'P231','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(13,'P241','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(14,'P251','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(15,'P261','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(16,'P271','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(17,'P281','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(18,'P291','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(19,'P222','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(20,'P223','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(21,'P224','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(22,'P225','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(23,'P226','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(24,'P127','Black Gown','Black Night Gown','PRG','Black','Black Night Gown'),

(25,'P128','Black Gown','Black Night Gown','PRG','Black','Black Night Gown');

To run the Java backend app, execute the ProductAppApplication.java file. Here are the endpoints for the Rest API :

Java Backend run @ http://localhost:8080

GET <http://localhost:8080/api/products> -- Result : 200 OK

[

{

"id": 1,

"productcode": "P111",

"productname": "Green Gown",

"category": "Green Night Gown",

"brand": "PRG",

"type": "Green",

"productdesc": "Green Night Gown"

},

{

"id": 2,

"productcode": "P122",

"productname": "Red Gown",

"category": "Red Night Gown",

"brand": "PRG",

"type": "Red",

"productdesc": "Red Night Gown"

},

{

"id": 3,

"productcode": "P133",

"productname": "Black Gown",

"category": "Black Night Gown",

"brand": "PRG",

"type": "Black",

"productdesc": "Black Night Gown"

}

]

GET <http://localhost:8080/api/products/P111> -- Result : 200 OK

{

"id": 1,

"productcode": "P111",

"productname": "Green Gown",

"category": "Green Night Gown",

"brand": "PRG",

"type": "Green",

"productdesc": "Green Night Gown"

}

=======================================

POST http://localhost:8080/api/products

Body:

{

"productcode": "P555",

"productname": "Blue Gown",

"category": "Blue Night Gown",

"brand": "PRG",

"type": "Blue",

"productdesc": "Blue Night Gown"

}

Result : 201 Created

{

"id": 5,

"productcode": "P555",

"productname": "Blue Gown",

"category": "Blue Night Gown",

"brand": "PRG",

"type": "Blue",

"productdesc": "Blue Night Gown"

}

=======================================

PUT http://localhost:8080/api/products/P555

{

"productcode": "P555",

"productname": "Dark Blue Gown",

"category": "Dark Blue Night Gown",

"brand": "PRG",

"type": "Dark Blue",

"productdesc": "Dark Blue Night Gown"

}

Result : 200 OK

{

"id": 9,

"productcode": "P555",

"productname": "Dark Blue Gown",

"category": "Dark Blue Night Gown",

"brand": "PRG",

"type": "Dark Blue",

"productdesc": "Dark Blue Night Gown"

}

=========================================

DELETE http://localhost:8080/api/products/P555

Result : Deleted Product Code - P555