

SnappFood Backend Developer Internship Road Map

First, you will learn about [📺 Git Tutorial for Beginners: Learn Git in 1 Hour](#) and use it in all stages of the project.

In the **second** pre-step learning you should learn [📺 Database Tutorial for Beginners](#)
And the **third** pre-step is [📺 Data Structures and Algorithms for Beginners](#)

In all stages, you will also study theoretical topics while doing practical work. And after finishing each stage, you will evaluate the codes that you had written before with the new topics.

In the following order:

- Learn about internet topics
[📺 How does the internet work? \(Full Course\)](#)
- Learn about SOLID, DRY
- Read the book Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin)
- Read the book Object Design Style Guide (Matthias Noback)
- Read the book Advanced Web Application Architecture (Matthias Noback)
- Study [Design Pattern](#)
- Study the Symfony documentation
[Symfony](#)
- Learn about Docker The Ultimate Docker Course – Code With Mosh

[Ultimate Docker Course](#)

Problem 1: Basic Order Form

Create a simple PHP web page with an HTML form that collects the following information from the user:

Customer name

Email address

Product name

Quantity

Upon submission, display a confirmation message with the order details.

Problem 2: Adding Validation

Enhance the order form by incorporating basic validation:

Ensure that all fields are filled out.

Validate the email address format.

Ensure the quantity is a positive integer.

Display appropriate error messages if validation fails, and retain the entered data in the form.

Problem 3: Connecting to a Database

Integrate a MySQL database to store order information. Modify the PHP script to insert the order details into the database upon form submission.

Problem 4: Retrieving Orders

Create a new PHP page that retrieves and displays a list of orders from the database. Display order details such as customer name, email, product, and quantity in a tabular format.

Problem 5: Implementing User Authentication

Add a basic user authentication system. Users should register with a username and password. Modify the order form and retrieval pages to only allow authenticated users to place orders and view order history.

Problem 6: Enhancing Security

Implement security measures:

Use prepared statements to prevent SQL injection.

Hash and salt user passwords.

Implement session management for user authentication.

Problem 7: Implementing Product Categories

Extend the order form to include a dropdown for product categories. Modify the database structure to store products by category. Update the order retrieval page to allow filtering orders by category.

Problem 8: Integrating API for Payment

Incorporate a payment gateway API into the order form to handle payments securely. Update the database to track payment status.

Problem 9: Implementing Order Status

Extend the database schema to include an order status field. Implement functionality to update and display the order status (e.g., pending, shipped, delivered).

Problem 10: Building a RESTful API

Create a RESTful API that allows external systems to place orders and retrieve order information. Secure the API with token-based authentication.

Problem 11: Understanding autoload vs include or require in php

Problem 12:

Do all the previous steps that we did with raw php in the framework, Symfony (design Model , Controller , Routes , ..)

Learning about Composer.json

Learning About Dependency injection

Final Project :

In this project, we want to design an order-taking software for the Buteh restaurant.

In the **First phase**, This restaurant has the ability to **define a menu** and also has the possibility of **registering orders** for this restaurant.

In the **Second phase**, the possibility of **categorizing products, searching for products, defining discount codes** and **applying discount codes** in order registration.

In the **Third phase**, **branch** and **schedule** work for the restaurant should be defined.

*** Dockerizing the project is **mandatory** and using Git is **required**.

The topics that will be evaluated in this project are:

- Model and database design
- UseCase detection and rest api design
- Scalability of the written code
- Adherence to solid, dry, clean code principles
- Adherence to security principles of apis and databases