

# Functional Deliverables

## Today's Menu



Paya  
PKR 400



Karahi  
PKR 650



Haleem  
PKR 350



Aloo Keema  
PKR 450

Desktop



Paya  
PKR 400



Karahi  
PKR 650

Tablet

# Today's Menu



**Paya**

PKR 400

PKR 450



**Mobile**



## Paya

~~PKR 450~~ PKR 400

★★★★★ | 8 Customer Reviews

**Paya Curry** is prepared with a base of onions, garlic, ginger, and a blend of aromatic spices such as turmeric, cumin, coriander, and garam masala, giving it a robust and earthy flavor. The trotters are simmered for hours, allowing the marrow and collagen to infuse into the curry, creating a silky, gelatin-rich consistency.

[Order Now](#)

Units in Stock : 15

Category : Food

[Description](#)

[Additional Information](#)

[Reviews \[8\]](#)

**Paya Curry** is prepared with a base of onions, garlic, ginger, and a blend of aromatic spices such as turmeric, cumin, coriander, and garam masala, giving it a robust and earthy flavor. The trotters are simmered for hours, allowing the marrow and collagen to infuse into the curry, creating a silky, gelatin-rich consistency.

## Desktop



## Paya

~~PKR 450~~ PKR 400

★ ★ ★ ★ ★ | 8 Customer Reviews

**Paya Curry** is prepared with a base of onions, garlic, ginger, and a blend of aromatic spices such as turmeric, cumin, coriander, and garam masala, giving it a robust and earthy flavor. The trotters are simmered for hours, allowing the marrow and collagen to infuse into the curry, creating a silky, gelatin-rich consistency.

[Order Now](#)



Haleem

Rs. 400

1

Rs. 400



Karahi

Rs. 700

1

Rs. 700



### Cart Totals

Subtotal

Rs. 1100

Total

**Rs. 1100**

Checkout

**QuickFood.**

Links

Help

Shahrah-e-Faisal Block 6  
PECHS,  
Karachi, Sindh

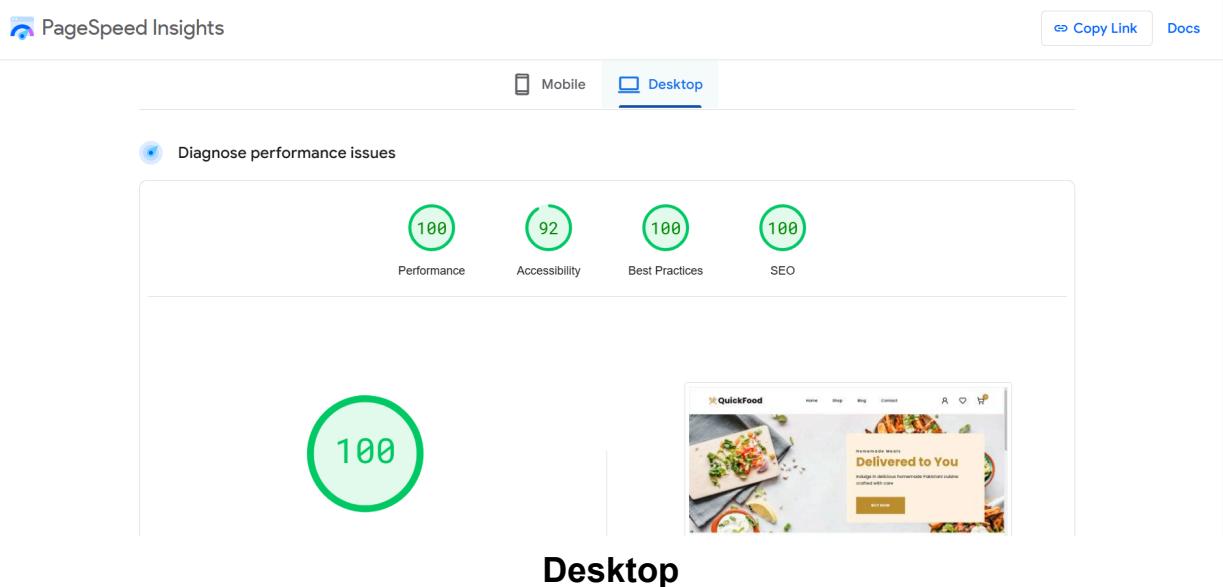
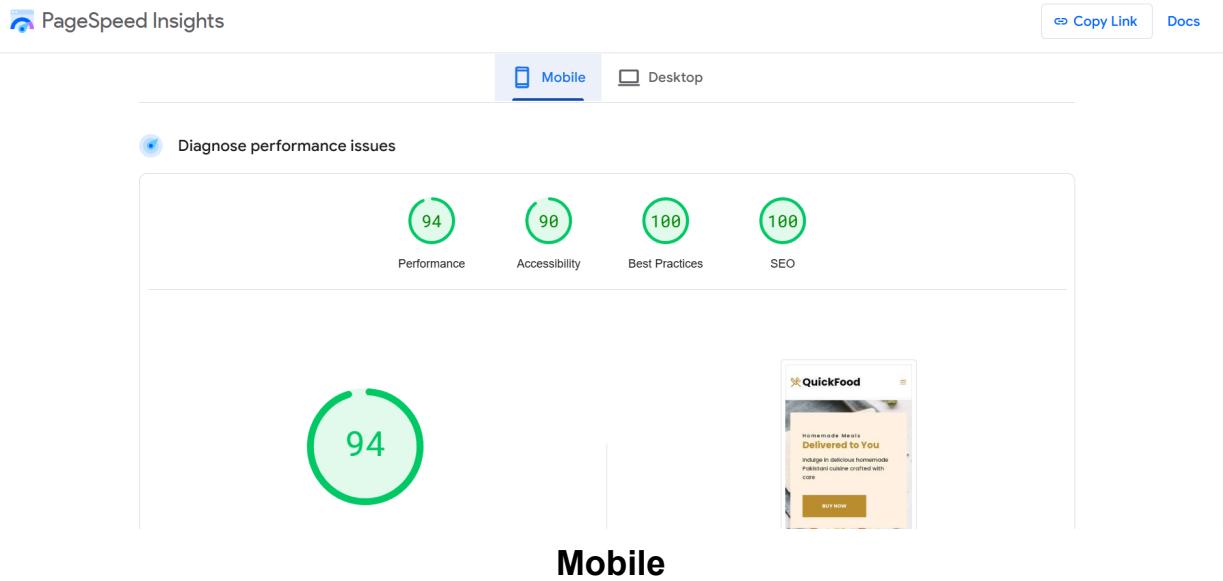
[Home](#)

[Payment Options](#)

[Shop](#)

[Returns](#)

**Speed Test**



## API Test

---

The screenshot shows the Postman interface with a collection named "Tickets Please". A test case titled "GET Get Products" is selected. The request URL is `http://localhost:3000/api/products`. The response status is 200 OK, with a response time of 725 ms and a size of 8.1 KB. The response body is a JSON object representing a product:

```
1 [  
2   {  
3     "_rev": "eW4V35ClPqhGTVkDD$3n",  
4     "_type": "product",  
5     "title": "Paya",  
6     "price": 480,  
7     "_createdAt": "2025-01-18T07:43:55Z",  
8     "slug": {  
9       "current": "paya"  
10    }  
11  }  
12 ]
```

The screenshot shows the Postman interface with the same collection "Tickets Please". A test case titled "GET Get a Product" is selected. The request URL is `http://localhost:3000/api/products/biryani`. The response status is 200 OK, with a response time of 26.43 s and a size of 807 B. The response body is a JSON object representing a product with an image asset:

```
1 {  
2   "image": {  
3     "_type": "image",  
4     "asset": {  
5       "_ref": "image-c91ecbc74b01291fda1edf6373183ea9515aff07-1200x1201-jpg",  
6       "_type": "reference"  
7     }  
8   },  
9   "_rev": "eW4V35ClPqhGTVkDEQUL",  
10 }
```

## Test Cases Executed and Their Results

## **Summary:**

Total Test Cases: 20

Passed Test Cases: 18

Failed Test Cases: 2 (Filter is not working, Pagination not working)

## **Performance Optimization Steps Taken**

---

To ensure optimal performance in the Next.js application, the following steps were taken:

### **Image Optimization:**

- **TinyJPG:** Used TinyJPG for compressing and optimizing images without losing quality. This service reduces the file size significantly, improving page load times while maintaining image clarity.
- **Image Resizing:** Implemented image resizing to serve images in the appropriate dimensions based on the user's device. This reduces unnecessary data usage, especially for users with smaller screens or lower resolution displays.
- **Next.js Image Optimization:** Leveraged Next.js's built-in `next/image` component to serve images in modern formats (such as WebP and AVIF) depending on the browser. This further improves loading speed and reduces bandwidth usage.

### **Lazy Loading:**

- Implemented lazy loading for non-essential content, such as product images, which ensures they are only loaded when they enter the viewport.

### **Server-Side Rendering (SSR):**

- Used SSR for pages that require fast initial loading times, enhancing performance and SEO.

## **Security Measures Implemented**

---

To safeguard the application and ensure secure handling of sensitive data, the following security measures were implemented

### **Sensitive Data in `.env`:**

- Stored sensitive data like API keys and credentials in `.env` files, ensuring they are not hardcoded in the codebase or exposed in version control.

### **Input Validation and Sanitization:**

- Implemented validation and sanitization of user inputs, particularly for search and filter fields, to prevent injection attacks and ensure data integrity.

### **Environment Variables Protection:**

- Secured sensitive API keys and authentication credentials in the `.env` file, which is excluded from version control.