

# Project Requirements



Brief overview of the project requirements.

## Docker environment

This project requires setting up a Laravel application in a Docker environment. The Docker environment will consist of three containers:

### Backend Container

This container will host the Laravel application, serving as the **backend** for our project.

### Frontend Container

This container will host the frontend of our application. It will interact with the backend container to fetch and display data.

### Database Container

This container will host the database for our application. It will interact with the backend container to store and retrieve data.

---

## Laravel Installation

Laravel and its dependencies will be installed in the backend container. This will allow the Laravel application to run in an isolated environment, ensuring that it has all the **necessary dependencies** without affecting the host system.

---

## Docker Network Configuration

All three containers will be configured to use the same network interface. This will allow them to **interact with each other**, as they will be able to see each other and communicate. This is crucial for the functioning of the application, as the frontend needs to fetch data from the backend, and the backend needs to store and retrieve data from the database.

---

## Running the Project

To run the project, we will use `docker-compose up` to bring up all three containers simultaneously. The network interface configurations will be specified in the Docker Compose file.

docker-compose up

---

## Docker Containers Configuration

This section outlines the detailed instructions for setting up each Docker container for the project.

### Backend

```
In [ ]: #!/bin/bash

# Pull the latest Laravel Docker image
docker pull laravelphp

# Create a new Docker container for the backend
docker run -d --name vistaqs_backend -p 8000:80 laravelphp

# Verify that the container is running
docker ps -a
```

### Frontend

```
In [ ]: #!/bin/bash

# Pull the latest Nginx Docker image
docker pull nginx

# Create a new Docker container for the frontend
docker run -d --name vistaqs_frontend -p 8080:80 nginx

# Verify that the container is running
docker ps -a
```

### Database

```
In [ ]: #!/bin/bash

# Pull the latest MySQL Docker image
docker pull mysql

# Create a new Docker container for the database
docker run -d --name vistaqs_database -e MYSQL_ROOT_PASSWORD=my-secret-pw -p 3306:3306 mysql

# Verify that the container is running
docker ps -a
```

---

## Laravel Installation in Docker Container

This section provides a comprehensive script detailing the steps required to install Laravel within a Docker container.

```
In [ ]: #!/bin/bash

# Access the backend container
docker exec -it vistaqs_backend bash

# Once inside the container, install Composer, a dependency manager for PHP
curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin

# Navigate to the project's directory
cd /vistaqs/main

# Install the dependencies
composer install

# Change the ownership of the project to the www-data user
chown -R www-data:www-data /var/www

# Exit the container
exit
```

## Configuring Network Interface for Docker Containers

Provide instructions on how to configure the network interface so that the Docker containers can interact with each other.

```
In [ ]: #!/bin/bash

# Create a new Docker network
docker network create vistaqs_network

# Connect the backend container to the network
docker network connect vistaqs_network vistaqs_backend

# Connect the frontend container to the network
docker network connect vistaqs_network vistaqs_frontend

# Connect the database container to the network
docker network connect vistaqs_network vistaqs_database

# Verify the network configuration
docker network inspect laravel_network
```

## Running the Docker Containers

docker-compose.yml file:

```
In [ ]: version: '3'
services:
```

```
backend:
  image: laravelphp
  container_name: vistaqs_backend
  ports:
    - 8000:80
  volumes:
    - ./:/var/www
  depends_on:
    - database
  networks:
    - vistaQs_network

frontend:
  image: nginx
  container_name: vistaqs_frontend
  ports:
    - 8080:80
  volumes:
    - ./nginx/default.conf:/etc/nginx/conf.d/default.conf
    - ./:/var/www
  depends_on:
    - backend
  networks:
    - vistaQs_network

database:
  image: mysql
  container_name: vistaqs_database
  volumes:
    - db_data:/var/lib/mysql
  environment:
    MYSQL_ROOT_PASSWORD: ${DB_PASSWORD}
    MYSQL_DATABASE: ${DB_DATABASE}
    MYSQL_USER: ${DB_USERNAME}
    MYSQL_PASSWORD: ${DB_PASSWORD}
  networks:
    - vistaqs_network

networks:
  vistaQs_network:
    driver: bridge

volumes:
  db_data:
```

```
In [ ]: # To start all the services defined in the `docker-compose.yml` file
        docker-compose up

        # To start all the services in the background (detached mode)
        docker-compose up -d

        # To stop the services
        docker-compose down

        # To view the status of the services
        docker-compose ps

        # To view the logs of the services
        docker-compose logs
```

