

# Setup Guides

This the guides step by step to run the Surf-Logistics website.

## Prerequisites:

1. Node.js and npm (or yarn)
  2. MongoDB Database
- 

Open a terminal or command prompt and navigate to the root directory of the project using `cd`

## Installing Client Dependencies and Building the Project

### 1. Integrating Tawk.to Live Chat with Your Website (Including Multiple Languages)

This guide walks you through integrating Tawk.to live chat with your website, supporting multiple languages with separate widget IDs.

## Prerequisites:

- A Tawk.to account (free tier available).

## Steps:

### 1. Create a Tawk.to Account and Property:

1. Visit <https://www.tawk.to/> and create a free Tawk.to account.
2. Follow the steps to create a new property for your website.
3. Make note of the **Property ID** displayed in the dashboard. You'll need it later.

## 2. Add Multiple Language Widget IDs:

1. Within your Tawk.to property dashboard, navigate to the **"Settings"** section.
2. Go to the **"Chat Widgets"** tab.
3. Here, you'll see the default widget ID for your property. This will be used for the main language on your website.
4. To add a widget for another language, click on **"Add Widget"**.
5. Customize the widget settings for the specific language. For example, you could configure the widget language and pre-defined messages.
6. Once finished, note down the unique **Widget ID** for this language. Repeat this process for additional languages you want to support.

## 3. Set Up Environment Variables:

- Inside the `client/.env` file, edit the following lines, replacing placeholders with your actual

values :

```
VITE_APP_TAWK_PROPERTY_ID=YOUR_PROPE  
RTY_ID          # Replace with  
PROPERTY ID  
VITE_APP_TAWK_EN_WIDGET_ID=YOUR_ENGL  
ISH_WIDGET_ID  # Replace with  
English widget ID  
VITE_APP_TAWK_FR_WIDGET_ID=YOUR_FREN  
CH_WIDGET_ID   # Replace with French  
widget ID
```

- Do not forget to save the file

#### 4. Install Client Dependencies:

- Navigate to the 'client' directory in your root folder.
- Run the following command in your terminal :

```
npm install
```

This will install all the necessary dependencies for your React application.

#### 5. Build the Project:

- After installing dependencies, run the following command to build the project :

```
npm run build
```

This will create an optimized production-ready build of your React application in the `dist` folder.

# Installing Server Dependencies and Running the Server

## 1. Install Server Dependencies:

- Navigate to the 'server' directory in your root folder.
- **Setting Up Environment Variables:**

1. Create a new file named `.env` in the root directory of your server folder `./server`.

This file will store your environment variables:

- In your server folder, create a new file named `.env`.

## 2. Add environment variables:

- Paste the following content into the `.env` file, replacing the placeholders with your actual values :

```
MONGO_URL=mongodb://localhost:27017/  
final_surf
```

```
JWT_SECRET=IcGuR0dBKg5PAreaY9Sqz1VlQ  
w6yFjMv  
JWT_LIFETIME=30d  
NODE_ENV=production  
PORT=5000
```

### 3. Save and restart the server:

4. Make sure that the Mongo databases work on the outlet 27017 and if you want to adjust the location of the database, adjust the "MONGO\_URL" variable in the file "server/.env"

- Save the `.env` file and restart your Node.js server. The environment variables should now be accessible in your code.
- Run the following command in your terminal :

```
npm install
```

This will install the required dependencies for your Node.js server.

## 2. Run the Server:

- Once the server dependencies are installed, start the server using the following command :

```
npm start
```

This will start your Node.js server, listening on the specified port (usually `5000` by default, but check your `.env` file for the exact port).

### Important Notes:

- **Environment Variables:** Ensure you have set up your environment variables correctly as described in the previous guide.
- **Database Connection:** If your server uses a database (e.g., MongoDB), make sure it's running and properly configured.
- **Other Dependencies:** If your project has additional dependencies or configurations, follow the specific instructions provided in your project's documentation.

### Choose a Hosting Provider:

- To host the website you can choose any hosting provider can running Nodejs apps, like :
  1. Netlify
  2. Render
  3. Heroku
  4. Hostinger (VPS)
  5. Vercel