

COVER PAGE

Deployment Instructions & Administration Manual



Autoaid Smart Repair station management System

Group 28

3rd year group project

Table of Contents

Table of Contents

Draw

Design

Layout

References

Mailings

Review

View

Help

Picture Format

Table

Pictures

Shapes

Icons

3D Models

SmartArt

Chart

Screenshot

Reuse Files

Get Add-ins

My Add-ins

Wikipedia

Online Videos

Link

Bookmark

Links

ables

Illustrations

Reuse Files

Add-ins

Media

Links

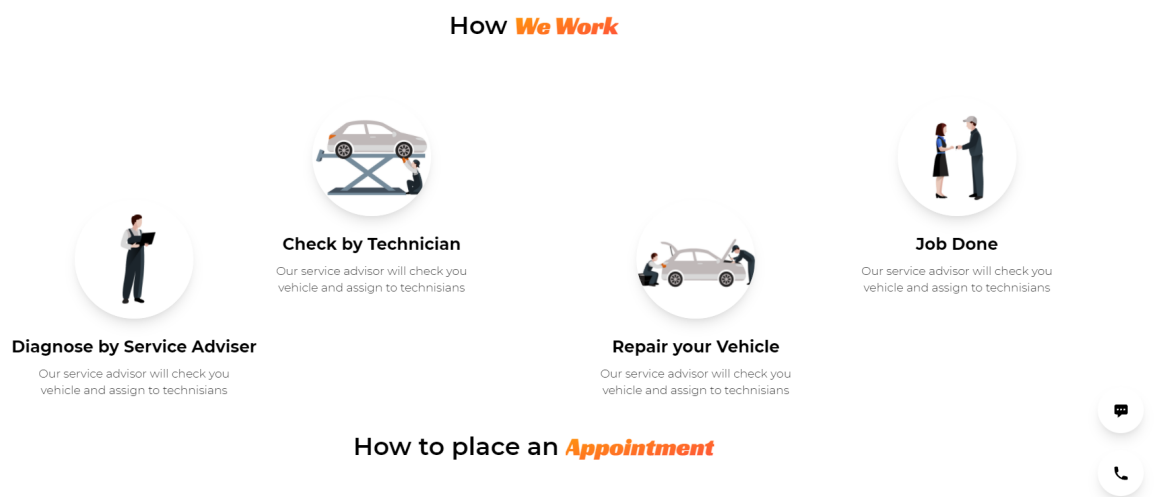
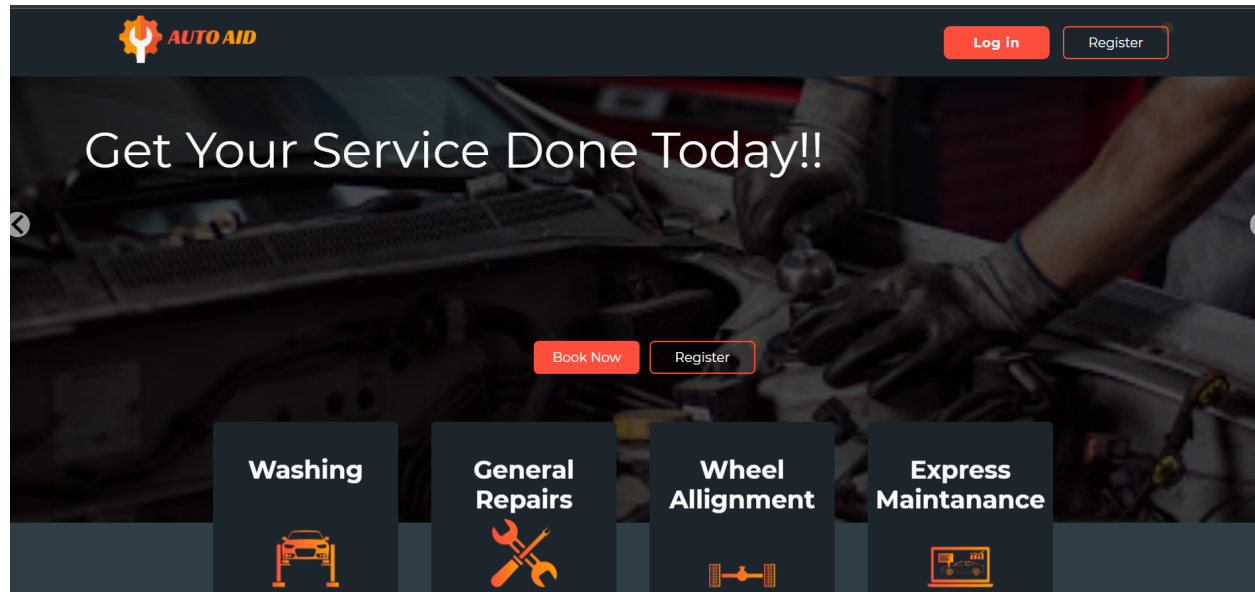
Smart Repair and Service Station Management System

User Manual

Table of Content

Introduction to the Autoaid Application

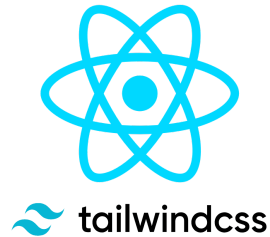
AutoAid is a repair and service station management system which helps to provide efficient and reliable service to customers. By this, we provide smart systems for technicians, customers, and other staff of the repair and service station. AutoAid mainly consists of the appointment module, service management module, inventory handling module, staff handling module, finance and invoice module, and customer management module.



The demo is available at - <http://autoaid.xyz/>

Deployment Instructions

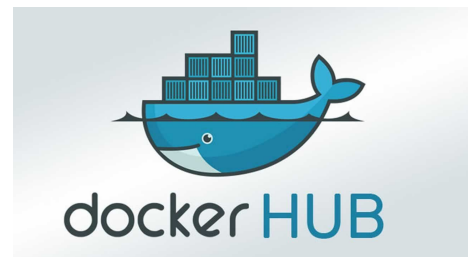
The Autoaid application is developed with the use technologies React js with Tailwind CSS for the front end and Spring Boot for the backend.



GitHub Link for Front End - <https://github.com/dumidu1998/autoaid-front>

GitHub Link for Back End - <https://github.com/dumidu1998/autoaid-API>

Both React application and the Spring boot application are dockerized with the help of Dockerfile and every push to the main branch will create a new image of each application and push to the Docker Hub with the use of GitHub Actions.



Docker Hub link for Front End https://hub.docker.com/r/dumidu1998/autoaid_front

Docker Hub link for Back End https://hub.docker.com/r/dumidu1998/autoaid_api

Setting Up for Development

Requirements

- React Js
- Java JDK 14
- IntelliJ Ultimate
- Mysql Server

Setting up for Front end Development

After the above requirements have been met:

1. Clone the repository and cd into it

```
git clone https://github.com/dumidu1998/autoaid-front.git  
cd autoaid-front
```

2. Install Dependencies

```
npm install
```

3. Setup Environmental Variables

Replace the URL in the declaration `REACT_APP_API_BASE_URL` with the URL of the Backend Endpoint

Eg = REACT_APP_API_BASE_URL=http://143.110.247.156:9009

4. Run the project

```
npm start
```

Setting up for Backend Development

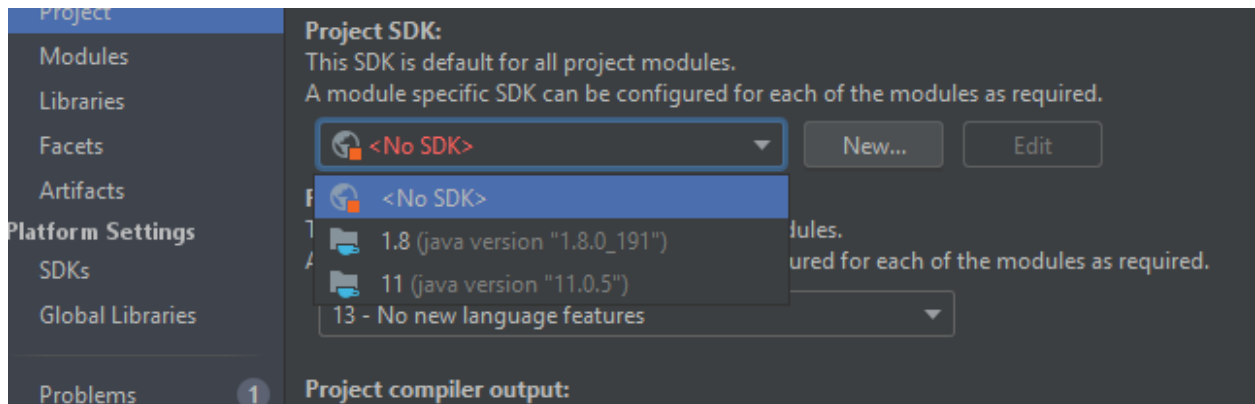
Clone the repository and cd into it

```
git clone https://github.com/dumidu1998/autoaid-front.git  
cd autoaid-front
```

Open the project folder with IntelliJ



Click on Add Configuration and Add new Spring Boot project configuration by selecting the java version



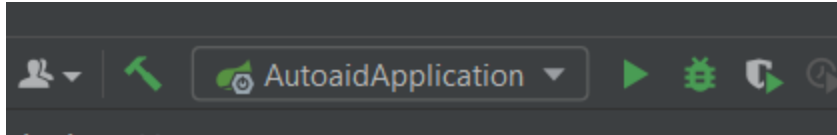
Then

Go to application.properties file and change the URL for the MySQL

```
spring.datasource.url = jdbc:mysql://localhost:3306/autoaid
```

```
spring.datasource.url = jdbc:mysql://localhost:3306/autoaid
```

Run the Application with Run button



Make sure to run the MySQL server in the development environment before starting the application.

If you need to change the port address of the backend, edit the application.properties file
`server.port = 9000`

Once All done user can access the front end with <http://localhost:3000>

And the backend with <http://localhost:9000>

Deployment of the production Build

For the deployment of the production build, you need to have the below requirement

- VPS in DigitalOcean(Recommended) or any other service Provider

Minimum requirement

VPS with 4GB ram and 2 CPUs

\$20 /mo \$0.030/hour
4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer

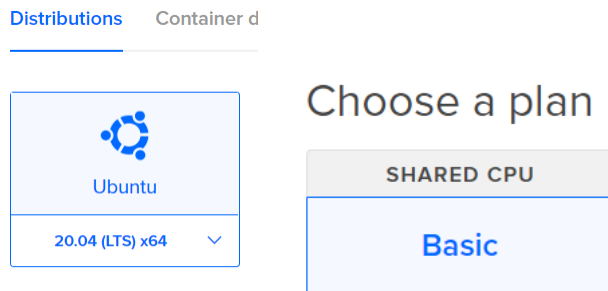
Recommended Requirement

VPS with 8 GB ram and 4 CPUs

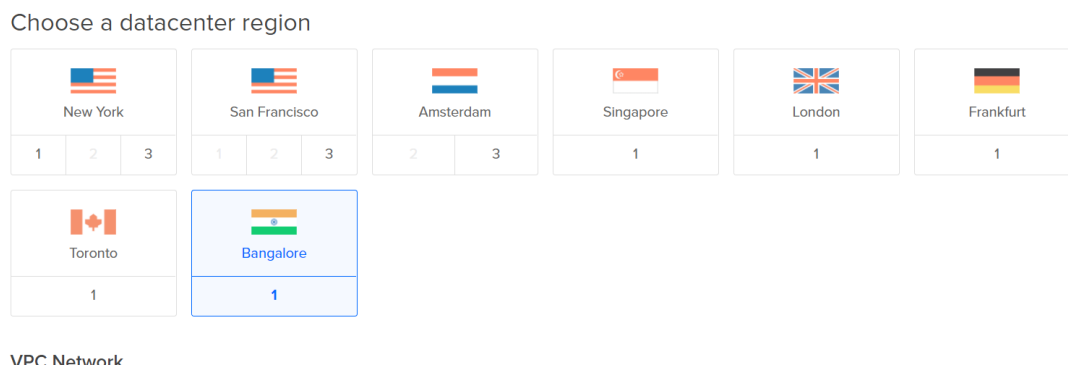
\$40 /mo \$0.060/hour
8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer

Steps for the deployment in the DigitalOcean VPS

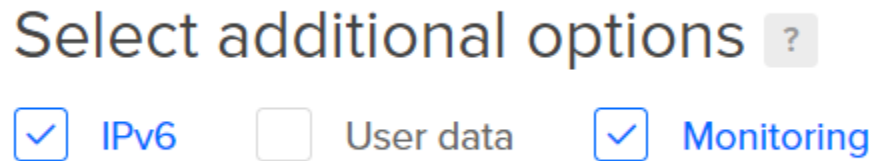
1. Login to the DigitalOcean account and open the form for creating a new Droplet
2. Choose the Distribution as ubuntu and VPS plan for Basic Shared CPU



3. Choose the datacenter region which is more closer to your location. (For Sri Lanka, Bangalore is ideal)

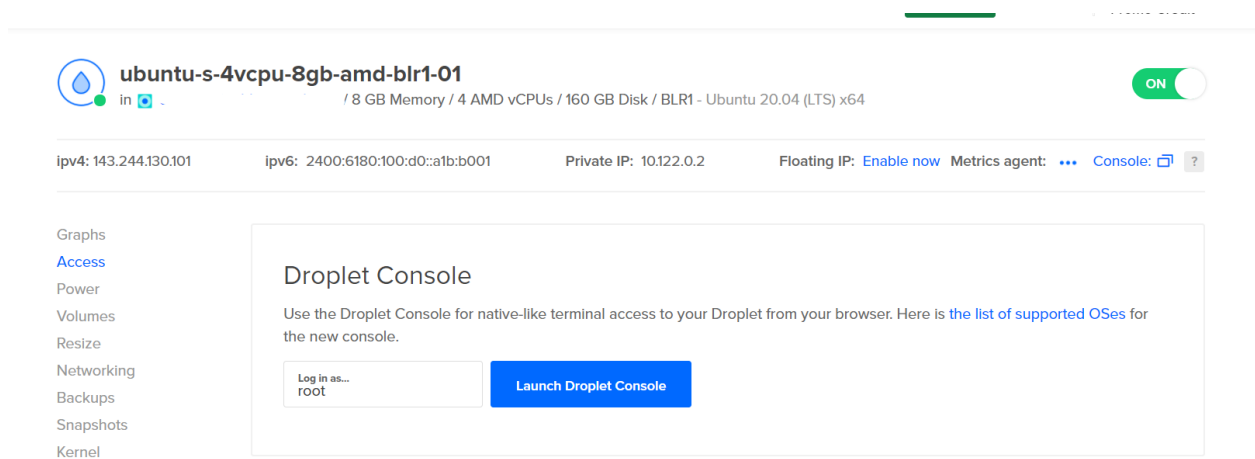


4. Select the Below Additional options



5. Then Create the Droplet

6. Once done open the Droplet console or Access the Console with SSH



There is a bash Script written to automate the Docker installation including the docker containers MySQL, PHPMyadmin, Autoaid frontend and the backend. Once completed the application is up and running on the server.

7. For that you need to first clone the repository and cd to it

```
git clone https://github.com/dumidu1998/autoaid-front.git
cd autoaid-front
```

8. Then the bash file for installing the application is here and you just need to run the bash file

```
bash install.sh
```

Once done you can access the application with the IP address followed by the port number 3000

Eg

<http://143.244.130.101:3000/>

To access the database using PHPMyAdmin visit your IP address with port 8899

Eg

<http://143.244.130.101:8899/>

The install script also installed Portainer.io which provides a simpler UI to manage the docker container. To access the Portainer, visit your IP address with port 9000

Eg

<http://143.244.130.101:9000/>

Band End is running on port 9009

<http://143.244.130.101:9009/> - backend

Below login Credentials are default credentials and you can use them to access the System

There are mainly 6 actors in the system and they have different dashboards.

Actor	Email	Username	Password
Customer	customer@gmail.com	Customer	customer123
Admin	admin@gmail.com	Admin	Staff123
Technician	technician@gmail.com	Technician	Staff123
Service Advisor	advisor@gmail.com	ServiceAdvisor	Staff123
Stock Keeper	skeeper@gmail.com	StockKeeper	Staff123
Cashier	cashier@gmail.com	Cashier	Staff123

If you need to set up the application for your own domain name, follow the instructions below.

Install Nginx

```
sudo apt install nginx
```

Create a file with your domain name inside the `/etc/nginx/sites-available` directory

Add following content to it

```
server {  
    listen 80;  
    listen [::]:80;  
  
    root /var/www/acc;  
    index index.html index.htm index.nginx-debian.html;  
  
    server_name autoaid.com www.autoaid.com;  
  
    location / {  
        proxy_pass http://localhost:3000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
}
```

Go to `/etc/nginx/sites-enabled` and run below command

```
sudo ln -s /etc/nginx/sites-available/autoaid.com /etc/nginx/sites-enabled/
```

Restart the nginx server

```
sudo systemctl reload nginx
```

System Administration Guide

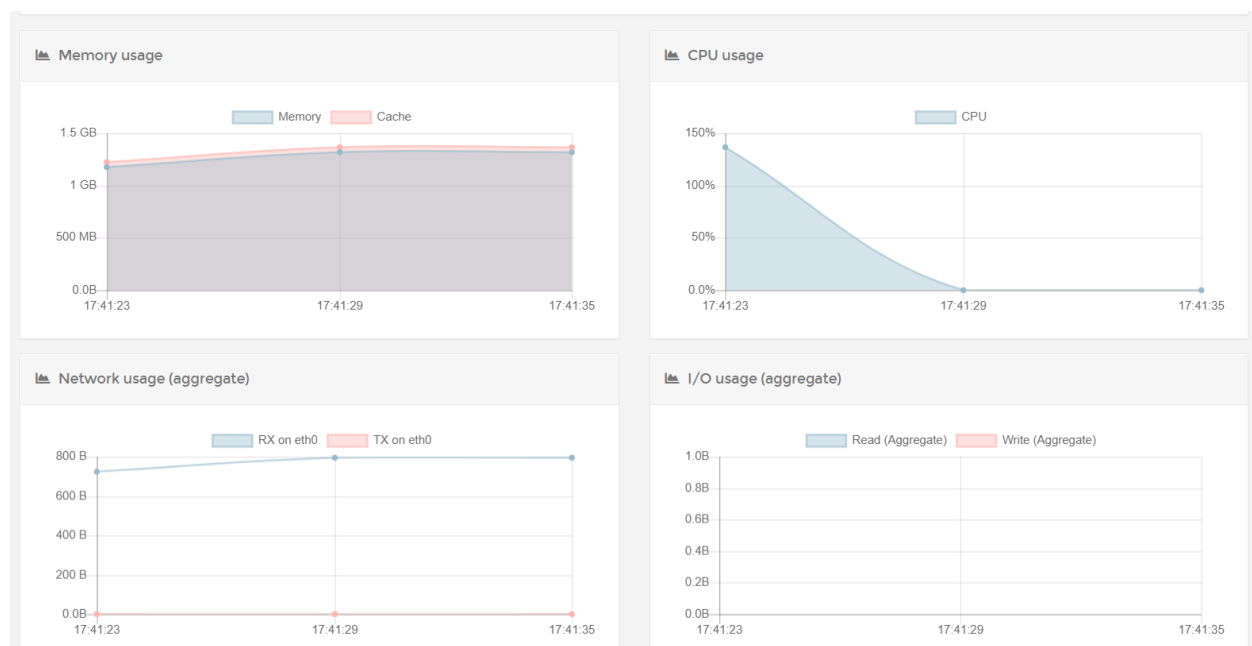
For managing the future versions and managing the containers access the Portainer console with the IP address with port 9000.

Eg - <http://143.244.130.101:9000/>

Signup and navigate to containers.

Name	State	Quick actions	Stack	Image	Created	IP Address	Published Ports	Owner
front	running	[Icons]	-	19970521/kitulgala-front:latest	2021-09-24 17:10:41	172.17.0.6	3000:3000 3000:3000	adm
api	running	[Icons]	-	19970521/kitulgala-api:latest	2021-09-24 15:49:53	172.17.0.5	3030:3030 3030:3030	adm
myadmin	running	[Icons]	-	phpmyadmin/phpmyadmin	2021-09-24 15:48:52	172.17.0.4	8899:80 8899:80	adm
api/mariadb	running	[Icons]	-	mariadb	2021-09-24 15:47:03	172.17.0.3	-	adm
portainer	running	[Icons]	-	portainer/portainer-ce	2021-09-24 15:46:30	172.17.0.2	8000:8000 8000:8000 9000:9000 9000:9000	adm

There you can see all the running docker containers in the VPS. You can see the status of them as well as logs and also you can attach to the containers and use command-line.



If you have updated the source code and you need to make those changes to the production, follow the below steps.

Click on the container and go to its page

Actions

▶ Start

■ Stop

● Kill

↺ Restart

⏸ Pause

▶ Resume

🗑 Remove

🔄 Recreate

📄 Duplicate/Edit

Container status

ID	b78dff3c51773c6fd60b04bcbd45f5b95234e5e72d50731da29d7c10743e741
Name	front
IP address	172.17.0.6
Status	Running for 2 minutes
Created	2021-09-24 17:40:22
Start time	2021-09-24 17:40:23

Logs Inspect Stats Console Attach

Click on Recreate

front

Are you sure?

You're about to re-create this container, any non-persisted data will be lost. This container will be removed and another one will be created using the same configuration.

Pull latest image ☒

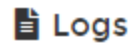
Cancel

Recreate

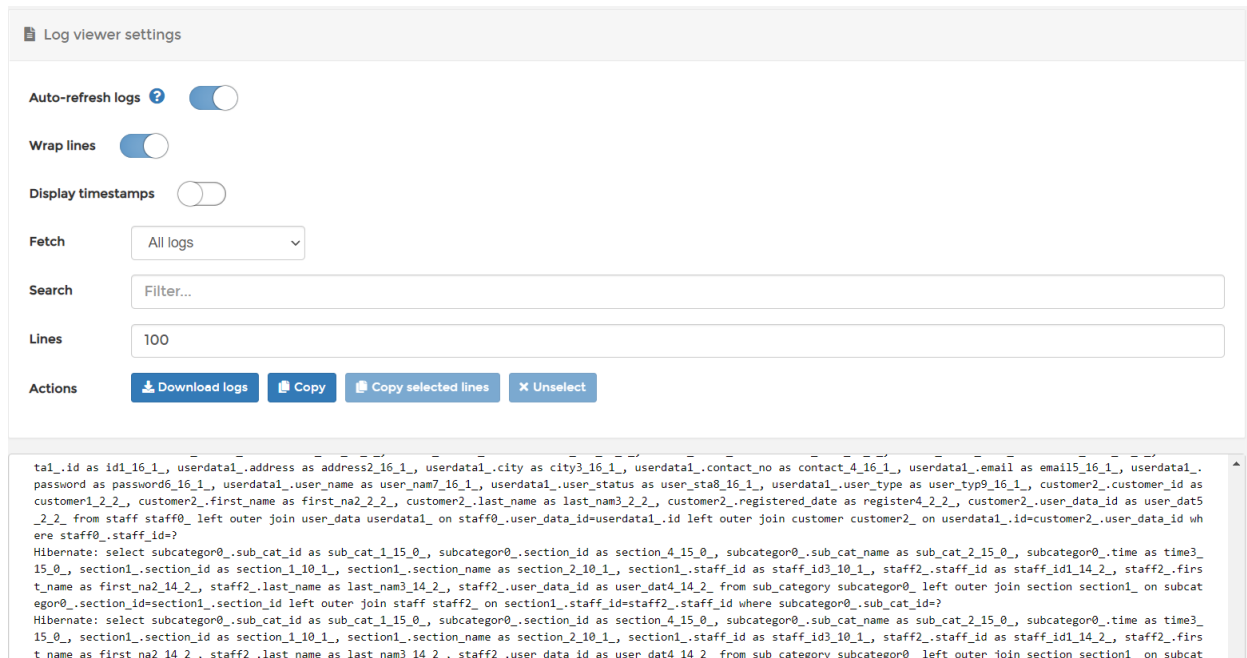
Enable 'pull latest image' and click recreate. After few seconds, the changes will apply!!

View logs in the spring boot.

Navigate to the container and click on the logs icons



Then the user will be redirected to the page which shows the logs from the date of start.



User can read them, search them or download the logs as a txt file.

Accessing the Database

For easy access to the database, Phpmyadmin is also installed.

To access Phpmyadmin, visit port 8899 followed by the IP address

Eg - <http://143.244.130.101:8899/>

Default username and passwords are

Username - root

Password - root

Once logged navigate to the autoaid database with the left side navigation panel.

Recent Favorites

New

autoid

New

appointment

appointment_slot

customer

hibernate_sequence

inventory_item

invoice

item_add

item_category

item_request

rate_and_review

repair

section

service_entry

slot

special_item_request

staff

sub_category

user_data

vehicle

information_schema

mysql

Table	Action	Rows	Type	Collation	Size	Overhead
appointment	Browse Structure Search Insert Empty Drop	7	MyISAM	utf8mb4_general_ci	5.3 KiB	-
appointment_slot	Browse Structure Search Insert Empty Drop	13	MyISAM	utf8mb4_general_ci	2.3 KiB	-
customer	Browse Structure Search Insert Empty Drop	8	MyISAM	utf8mb4_general_ci	3.3 KiB	-
hibernate_sequence	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8mb4_general_ci	1.0 KiB	-
inventory_item	Browse Structure Search Insert Empty Drop	27	MyISAM	utf8mb4_general_ci	5.7 KiB	-
invoice	Browse Structure Search Insert Empty Drop	6	MyISAM	utf8mb4_general_ci	3.2 KiB	-
item_add	Browse Structure Search Insert Empty Drop	29	MyISAM	utf8mb4_general_ci	3.9 KiB	-
item_category	Browse Structure Search Insert Empty Drop	4	MyISAM	utf8mb4_general_ci	3.1 KiB	-
item_request	Browse Structure Search Insert Empty Drop	8	MyISAM	utf8mb4_general_ci	5.5 KiB	-
rate_and_review	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8mb4_general_ci	1.0 KiB	-
repair	Browse Structure Search Insert Empty Drop	20	MyISAM	utf8mb4_general_ci	5.2 KiB	-
section	Browse Structure Search Insert Empty Drop	5	MyISAM	utf8mb4_general_ci	3.2 KiB	-
service_entry	Browse Structure Search Insert Empty Drop	17	MyISAM	utf8mb4_general_ci	7.2 KiB	-
slot	Browse Structure Search Insert Empty Drop	18	MyISAM	utf8mb4_general_ci	3.7 KiB	20 B
special_item_request	Browse Structure Search Insert Empty Drop	3	MyISAM	utf8mb4_general_ci	4.2 KiB	-
staff	Browse Structure Search Insert Empty Drop	19	MyISAM	utf8mb4_general_ci	3.7 KiB	-
sub_category	Browse Structure Search Insert Empty Drop	15	MyISAM	utf8mb4_general_ci	3.6 KiB	-
user_data	Browse Structure Search Insert Empty Drop	27	MyISAM	utf8mb4_general_ci	9.3 KiB	-
vehicle	Browse Structure Search Insert Empty Drop	10	MyISAM	utf8mb4_general_ci	4.6 KiB	-
19 tables	Sum	237	InnoDB	utf8mb4_general_ci	78.8 KiB	20 B

↑

☐ Check all / Check tables having overhead

With selected:

▼

If the system admin needs to remove any users, the admin needs to remove the related entry from the user_data table as well as the respective table. Eg if the customer, remove it from the customer table. If staff, remove from the staff table.

* All the passwords are hashed and are cannot be reversed or modified.

After the above requirements have been met.

1. Clone this repository and cd into it

```
git clone https://github.com/Proktara/proktara-streamer.git
cd proktara-streamer
```

2. Install dependencies

```
yarn install
```