**Git Contribution Report**

**G27**

We created an organization in GitHub and created repositories in it. You can view the organization using the below link. Also, we sent an invitation to join the organization to [*y3projects@ucsc.cmb.ac.lk*](mailto:y3projects@ucsc.cmb.ac.lk) *email address*

**Organization**

Green-Core - <https://github.com/Green-core>

**Repositories**

Documents - [https://github.com/Green-core/Documents](https://github.com/Green-core/Documents.git)

* To handle designs and documentation of the system (All members contributed in design and documentation)

Mobile-App - <https://github.com/Green-core/Mobile-App>

* For the frontend of the mobile app

Mobile-App-Backend - <https://github.com/Green-core/Mobile-App-Backend>

* For the backend of the mobile app

Web-App - <https://github.com/Green-core/Web-App>

* For the frontend of the web app

Web-App-Backend - <https://github.com/Green-core/Web-App-Backend>

* For the backend of the web app

Arduino - <https://github.com/Green-core/Arduino>

* For the IoT device

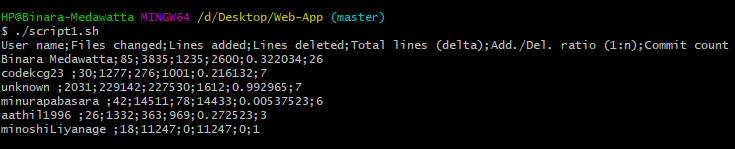
IoT-Device - <https://github.com/Green-core/IoT-Device>

* For the backend server of the IoT device

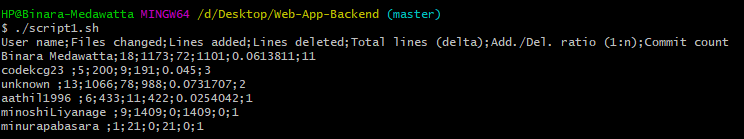
**Group Contribution (Script 1)**

**Web App**

**Repository - Web-App**

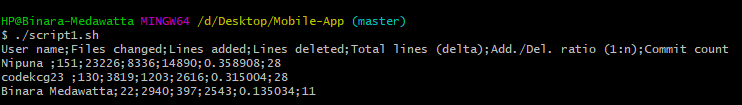
****

**Repository - Web-App-Backend**

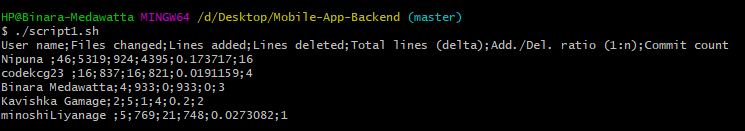
****

**Mobile App**

**Repository - Mobile-App**

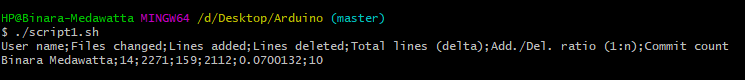
****

**Repository - Mobile-App-Backend**

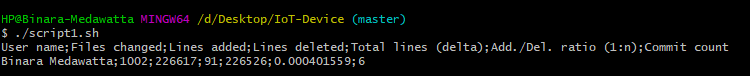
****

**IoT Device**

**Repository - Arduino**

****

**Repository - IoT-Device**

****

**Individual Contribution (Script 2)**

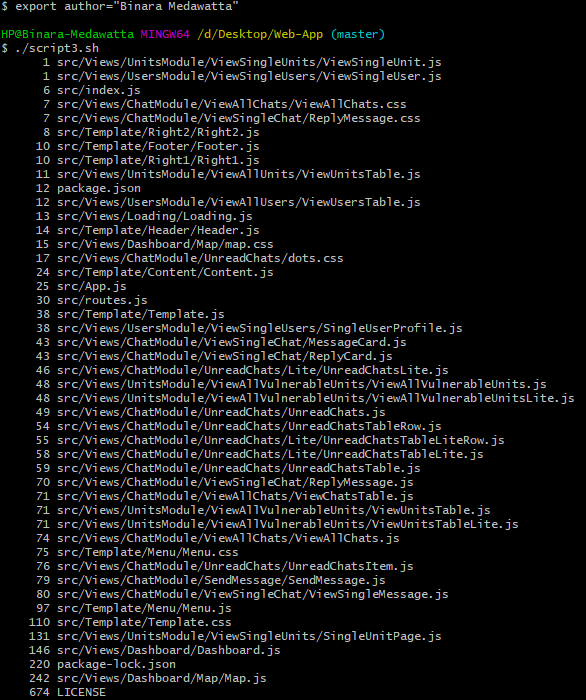
**B A Medawatta - 17001072 (Git username - Binara Medawatta)**

**Repository** - Web-App

**Branch** - Development

**Componentes**

* Chat Module for Admins
* Send Message - for admins to send messages to users regarding vulnerable units
* Reply Message - for admins to reply messages from users
* View All Messages - to view all the messages
* View Single Message - to view single message
* View New Messages - to view new (Unread) messages
* Navigation - Web App navigation
* The dashboard of the Web App
* Web App template

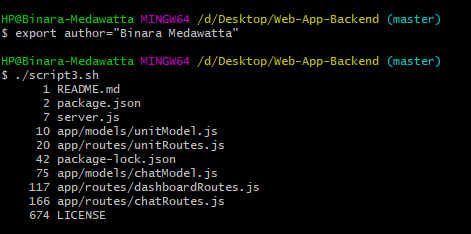


**Repository** - Web-App-Backend

**Branch** - Development

**Routes**

* /chats - API endpoints related to chat module
* /reply - API to reply message
* /send-message - API to send message
* /get-all - API to view all messages
* /get-one - API to view single message
* /get-unread - API to view unread messages
* /viewed-message - API to mark viewed messages
* /dashboard - API endpoints related to the Dashboard
* /get-total-users - API to get the total number of users
* /get-total-units - API to get the total number of units
* /get-total-vulnerable-units - API to get the total number of vulnerable units
* /get-total-unread-chats - API to get the total number of unread chats
* /get-district-unit-count - API to get the total units of a district for the Map
* /get-district-user-count - API to get the total users of a district for the Map
* /get-plant-count - API to get the total number of plants registered in the system

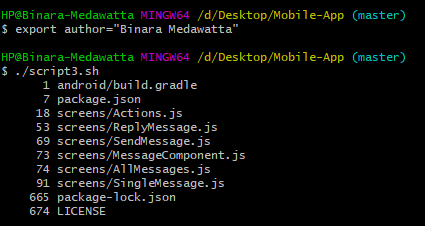


**Repository** - Mobile-App

**Branch** - dev

**Components**

* Chat Module for Users
* Send Message - for users to send messages to admins regarding issues with IoT units
* Reply Message - for users to reply messages from admins
* View All Messages - to view all the messages received and sent
* View Single Message - to view a single message

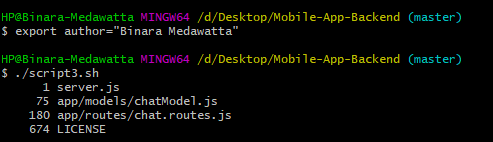


**Repository** - Mobile-App-Backend

**Branch** - dev

**Routes**

* /chats - API endpoints related to chat module
* /reply - API to reply message
* /send-message - API to send message
* /get-all - API to view all messages
* /get-one - API to view single message
* /get-unread - API to view unread messages
* /viewed-message - API to mark viewed messages

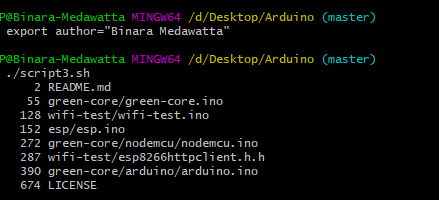


**Repository** - Arduino

**Branch** - master

**Components**

* Humidity detection
* Temperature detection
* Soil moisture detection
* Light intensity detection
* Bug detection
* Plant watering system
* Plant fertilizing system
* Grow lights control
* Bug repellent system



**Repository** - IoT-Device (Backend server of IoT Device)

**Branch** - master

**Routes**

* /units
* /connect-unit - to create a new IoT device document
* /update-data - to update sensor data
* /get-actuator-status - to get status of actuators (If in manual configuration the user switches on an actuator the IoT device should respond to it)



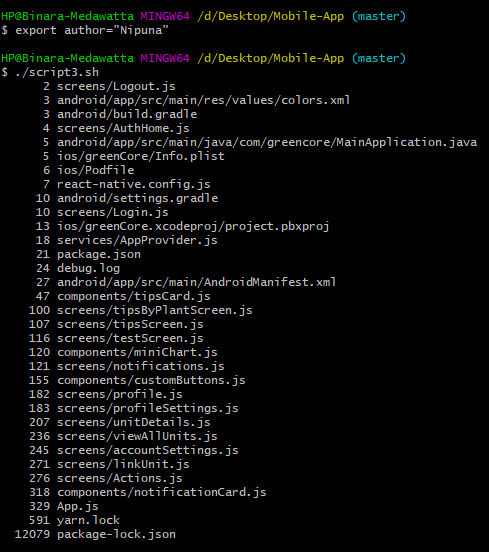
**J A N C Niroshana - 17001196 (Git username - Nipuna)**

**Repository** - Mobile-App

**Branch** - Dev/ Master

**Components**

* Units
* Link units - Helps users to connect new IoT Devices
* View linked units - View all connected units owned by the user
* View Unit details - View more details about a unit
* Control actuators - Control actuators related to a single unit
* Account settings - Change the password of the user
* Profile settings - Change username and telephone number related to a user
* Notifications - View notifications related to a user
* Unit details - Details related to a single unit
* Actions - Control actuators related to a single unit
* Tips - View tips related to the plants own by the user



**Repository** - Mobile-App-Backend

**Branch** - Dev

**Routes**

* /units

- Check the existence of a specific unit

- Update unit

- Delete unit

- Get all units related to one user

- Get data related to one unit

- Turn actuators on/off

* /plants - Retrieve all plant types
* /users

- check existing password using ID

- change the current password of a user

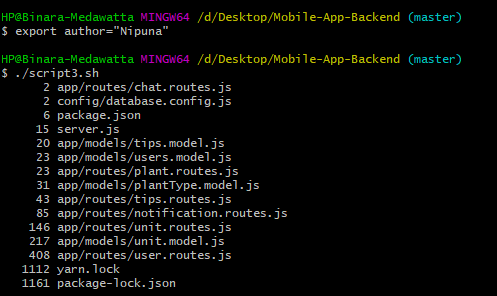
- change profile-related data of a user

* /tips

- Retrieve all plant types and id's of them related to a user

- Retrieve all tips related to a plant

* /notifications - API endpoint to receiving notifications



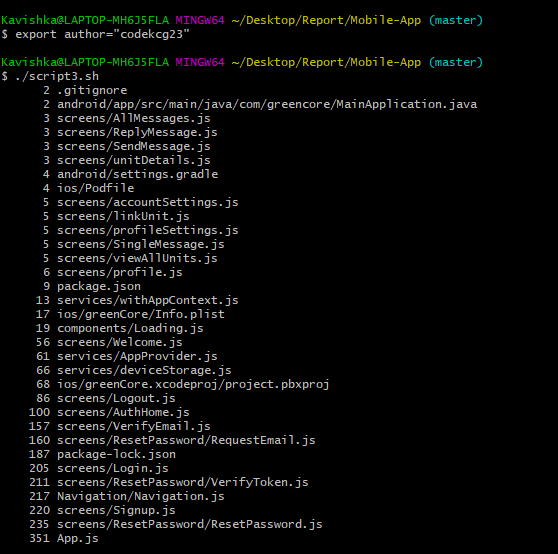
**K C Gamage - 17000475 (Git username - codekcg23)**

**Repository** - Mobile-App

**Branch** - dev

**Components**

* Splash screen – Every time the app opens this screen will show up
* Login
* User can log in with email and password
* Validation
* Receive JWT Token to secure all endpoints
* Persistent login with React context API and Asyncstorage
* Signup
* Register user
* Validation
* Email verification
* Navigation
* All components navigation
* Drawer and Stack navigation
* Forgot Password
* Request email screen
* Verify code screen
* Reset password screen

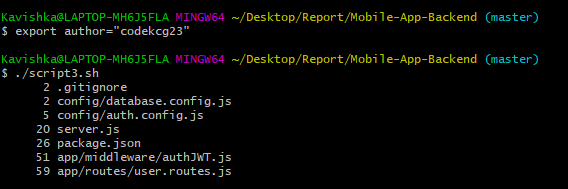


**Repository** - Mobile-App-Backend

**Branch** - dev

**Routes**

* /users/login
  + Check login credentials
  + Generate JWT Token
* /users/register
* Check signup fields
* Send email to users with verify code
* /users/forgotPassword
* Check email field
* Generate verify code
* Update user document with verification code and expiration time
* Send email to the user with verify code
* /users/checkToken
* Check verify code is valid or expired
* Update user document to stop another attempt
* /users/resetPassword
* Check password and Confirm Password field
* Update the user’s password

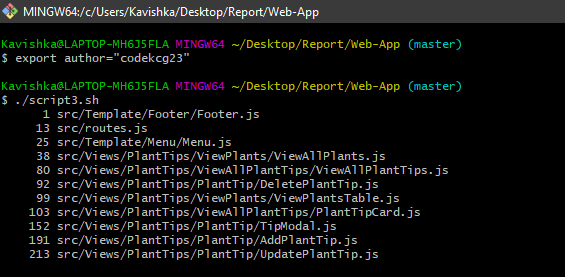


**Repository** - Web-App

**Branch** - Development

**Components**

* View All plants - View all the plants users has used with units
* View plant tips - View all plants tips related to plant type
* Update plant tip
* Delete plant tip

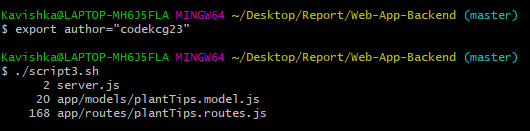


**Repository** - Web-App-Backend

**Branch** - Development

**Routes**

* /plants/get/
  + Retrieve all plants
* /plants/get/:id
* Retrieve individual plant details
* /plants/addTips/:id
* Add new plant tip to plant type
* /plants/updateTips/:id
* change existing plant tips
* /plants/removeTips/:id
* Remove plant tip from particular plant type



**H U K M Pabasara - 17020611 (Git username - minurapabasara)**

**Repository** - Web-App

**Branch** - Minura

**Components**

* Admin module
* View admin details component

Admins can see their own details

* View all admins component

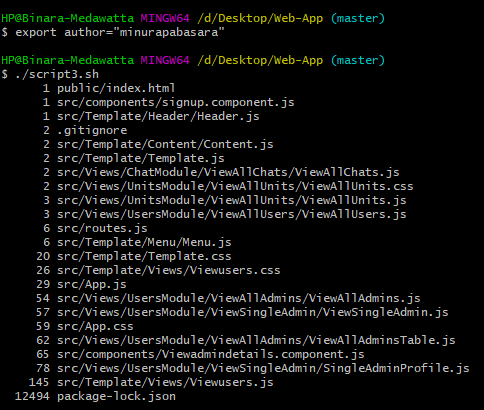
Admins can see all admins that are registered

* View Single admins component

Admins can see single Admin profiles that are registered

* Unit module
* View Vulnerable units component

Admin can see the units which are at risk

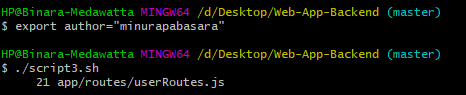


**Repository** - Web-App-Backend

**Branch** - minura\_dev

**Routes**

* /users
* /get/:id - API to view admin (currently logged in) details
* /get-all-admins - API to view all details of all admins in the system
* /get-single-admins - API to view details of a single admin in the system
* /units
* /view-vulnerable-units - API to view vulnerable units



**M.M.H Tharushika - 17001765 (Git username - unknown)**

**Repository** - Web-App

**Branch** - hasini

**Components**

* Signup Component
* Register User
* Validation
* Login Component
* User can login with email and password
* Forgot Password
* Verification
* Logout
* Validation

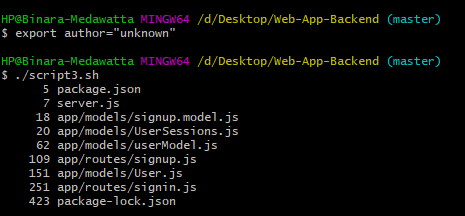


**Repository** - Web-App-Backend

**Branch** - hasini

**Routes**

* /signup - Api to register new users
* /register - API to save new user details.
* /signin - API to signin
* /get - Retrieve all the registered users.
* /login - API to login registered users.
* /verify - API to verify user credentials
* /forgotpassword - API to change forgot password
* /auth- Generate JWT token



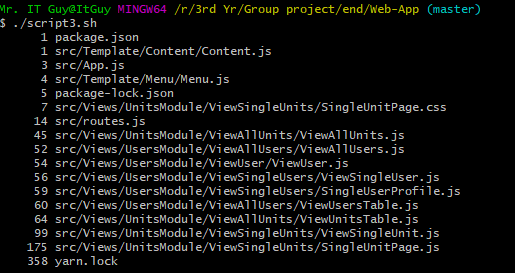
**K.S.A Ahamed - 17020034 (Git username - aathil1996)**

**Repository** - Web-App

**Branch** - aathil\_dev

**Components**

* User Module
* View all users - Admins are able to see all users in the system
* View single user - Admins are able to see the particular user’s details by clicking the “view” button.
* Unit module
* View all units - Admins are able to see all units in the system
* View single unit- Admins are able to see the particular unit’s details with respective graphs and charts by clicking the “view” button.
* Admin module
* Update admin details component - Admins can update their own details
* User interface designing (front end works)



**Repository** - Web-App-Backend

**Branch** - aathil\_dev

**Routes**

* /users
* /get-all - API to view all users’ details
* /get-one - API to view a particular user’s details
* /update/:id - API to update admin’s details
* /units
* /get-all - API to view all units’ details
* /get-one - API to view a particular unit’s details

