**Intelli-NICU Platform**

**Project Overview:**

Intelli-NICU (integrated Neonatal Intensive Care Unit) is an IoT based platform which automatically captures real-time clinical data from the diversified connected devices and send data to Cloud Platform.

**In Platform have 3 main components**

1. **Hardware kit**
2. **Android app (Ionic framework)**
3. **Software (Front-end & Back-end)**

**Hardware KIT & Android app (Ionic framework)**

Hardware KIT is basically a microcontroller which is built in Bluetooth BLE, Hardware KIT is connected with different sensor like Baby temperature, Air Temperature, Peripheral Temp, Delta(calculate from application level) , Set Temperature & Weight .

Hardware KIT is able to send data to android app every bit of minute , once they have pair & connected to each other.

**How hardware handshaking to application(Android app)?**

Hence application is staring, It first job is searching a MAC ID & connect with hardware kit, once they found & connect then it will send data to the application till the user disconnect the BLE or stop the machine.

**Ionic Framework:**

Ionic is an open-source UI toolkit for building cross-platform mobile, web, and desktop applications using web technologies such as HTML, CSS, and JavaScript/TypeScript(Angular)

Using ionic framework we can build android app as well as iOS app. With the single codebase we save the development time & effort.

Android App Prerequisite:

1. Bluetooth BLE 4 & 5
2. Ram Size : Minimum than 2 GB
3. Rom : Minimum 16 GB

Our android app is highly processing the incoming data string from the warmer retrofit, warmer retro kit pull the different type of string to android app , android app is getting that string in a form of encrypt format, after that will decrypt the receiving string & storing into separate tables.

**How data populating over the screen?**

Hence , we get data through BLE service then will emit that data to dashboard service , dashboard service is able to get that data & populate over the application dashboard.

The between data communication ,One more service playing the important role in the system, Service is storing the history data into the database over the APIS. The data storing interval is one minute .

Sample Data Format :

**T1,T2,T3,heater ,mode**

{"DID":"SBWXXX","MID":"201","AT":"42.8","BT":"30.3","PT":"33.4","HT":"0","M":"3"}

**hardware alerts**

{"DID": "SBWXXX","MID":"440","AL" : "['0', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0']"}

**Explain Data**  **accusation?**

In Data acquisition – mobile application will communicate with Device/Hardware over BLE and get the data string in bytes array. For each type of data, a separate string is created and which is distinguished using different MID. Based on MID data is extracted in Mobile application side and dumped. If Internet is available, then data is sync with Cloud else data string will ignore.

**Refer Document of data accusation for more information.**

**Software (Front-end & Back-end)**

The Overall Platform design & developed using MEAN Stack technology. For front-end we use Angular 8,HTML, CSS, JAVASCRIPT, Bootstrap , Typescript, Angular Material (UI Components).The deployment build create using ng build command & deploying source code using WinSCP software on AWS Ec2 Instance.

The Backend we developed using Node.js, All modules & database files are separately defined in the source code & maintaining those file using the Happi JS library.

Database Is crucial part of the system, which we use MYSQL.

**Logical diagram with architecture**

A diagram of a computer system

Description automatically generated

**Development Environment Setup:**

|  |  |  |
| --- | --- | --- |
| **Module** | **Technologies** | **Tools** |
| **Android App** | Ionic V3, Cordova, Java, Angular 8, Typescript, SQL-Lite | Android Studio, BLE Serial Monitor, Android Tab(testing) |
| **Platform Front-End** | Angular 8 , Typescript ,Bootstrap, Html , CSS , JavaScript , Angular Material , Chart js | VS Code |
| **Platform Back-End** | Node.js , JavaScript , Adobe library | VS Code, Postman, Browser |
| **Database** | My-SQL , SQL-Lite | MySQL Workbench |

**Installation Guide: Step-by-step instructions to set up the development environment.**

Installing Node.js and NPM.

Setting up MySQL.

Installing Angular CLI.

Cloning the project repository and installing dependencies.

**Configuration Files:**

**Platform Environments:**

**Production Environment:** Open for public domain (for all user & client)

URL : <https://intelli-nicu.com/>

**UAT Environment:** Open for Development Team & Client (product owner)

URL : <http://inframedtech-d.com/>

**Front-end build configuration:**

// The file contents for the current environment will overwrite these during build.

// The build system defaults to the dev environment which uses `environment.ts`, but if you do

// `ng build --env=prod` then `environment.prod.ts` will be used instead.

// The list of which env maps to which file can be found in `.angular-cli.json`.

// serverurl: "https://www.i-tems.in"; // Production Server URL

// serverurl: <http://3.109.33.14:3000> // UAT Server URL

// serverurl: <http://localhost:3000> // Local Server URL

export const environment = {

  production: false,

  serverurl: "https://www.i-tems.in"

};

**Back-end configuration file:**

CMS.json

{

    "apps": [{

        "name": "babywarmer\_backendapis",

        "script": "./server/server.js",

        "watch": true,

        "env": {

            "NODE\_ENV": "development"

        },

        "env\_production": {

            "NODE\_ENV": "production",

            "USERNAME": "root",

            "PASSWORD": "ganesh",

            "DATABASE": "babywarmerdb",

            "HOST": "localhost",

            "DIALECT": "mysql",

            "TIMEZONE": "+05:30",

            "APP\_SECRET": "BABYWARMER-secret-prodserver",

            "SERVERPORT": 8080,

            "SYSTEMKEY": "15419O105143556356",

            "SYSTEMID": 12,

            "COUNTRY": "india",

            "COUNTRYID": 1,

            "COUNTRYCODE": "+91",

            "ISOCODE": "IND",

            "ROUTS": "all",

            "ERRORLOGSOURCESERVER": "stage",

            "TOKENRANDOMCHARLENGTH": 3,

            "TOKENENCODETKEY": "#D^FG3@4as$df^5#",

            "APPSECRETUSER": "#M^LP3@4se$df^2#",

            "CLOUDSECRETUSER": "#Q^LP3@8le$df^8#",

            "CREATEORMODIFYBY": "backendapis@babywarmer.com",

            "OTPLENGTH": 6,

            "RECAPTCHASECRETKEY": "",

            "KEY": "123|a123123123123123@&"

        }

    }]

}

> Codebase Overview:  
  #Folder Structure: Explanation of the project’s folder structure.

1) Frontend

2) Backend

  #Key Modules and Components: Description of the main modules and components, their purpose, and interactions.

1. User master
2. Machine master
3. Site master
4. Site-Machine mapping
5. Client master & Control room manager
6. Control room manager Dashboard, Analytics, Alert Log, Doctor-panel, Nurse-panel, Discharge Report

#Third-Party Libraries: List of significant third-party libraries and their usage.

Back-end :

> Database Details  
  #Data Models: Explanation of data models used in the application.

  #Schema Design: Detailed description of database schemas and relationships.

Database backend configuration file: visit the backend code & demonstrate the configuration file

> API Documentation  
  #Endpoints: List of API endpoints with descriptions, request/response formats, and sample calls.

* We use swagger for API documentation.

#Authentication: Explanation of the authentication mechanism (e.g., JWT, OAuth).

* JWT

 #Error Handling: Common error codes and their meanings.

We create common function for locking the errors & storing inside folder

> Front-End Details  
  #Component Structure  
  #Routing: Explanation of the routing setup in Angular.  
  #State Management: Description of state management techniques used (e.g., NgRx, services).

> Deployment  
  # Build Process  
  # Deployment Instructions  
  # Server Configuration