



UNIVERSITY  
TECHNOLOGY,  
MAURITIUS

---

Name and Student Id:

*Mohun Mahima (2003\_19385)*

*Beeharry Mohammad Taajuddin (2003\_19375)*

Cohort:  
*BSE20AFT*

Module Name:

*Mobile Application Development*

Lecturer Name:  
*Dr. Geerish Suddul*

Date Submitted:  
*17/02/2022*

Assignment Title:  
*Pregnancy*

## Table of Contents

1.	Introduction.....	4
2.	Problems .....	4
3.	Scope.....	4
4.	Use Case.....	5
5.	Architecture of the proposed system.....	6
6.	Solution.....	7
➤	Functionalities that have been implemented: .....	7
a.	Use of jQuery Mobile UI features like panel, lists, navigation bar, and others .....	7
b.	Specific mobile related events like swipe, rotate, tap, and others.....	9
c.	Weight Tracker .....	10
d.	My Reminder .....	12
e.	Permanent Offline storage features.....	13
f.	Web service-related functionality ( <i>Contact Us</i> ) .....	15
g.	Splash Screen .....	17
h.	Geolocation services using the OSM Map Service ( <i>Location of Specialists</i> ).....	18
➤	Functionalities that have not been able implemented: .....	19
a.	Current Weather.....	19
b.	Kick Counter .....	20
7.	Testing.....	22
7.1	Homepage .....	22
7.2	Weekly Information (Homepage) .....	22
7.3	Pregnancy Timeline (Homepage) .....	23
7.4	My Bump (Homepage) .....	25
7.5	Ultrasound Image (Homepage) .....	25
7.6	Symptoms Page .....	26
7.7	Tools Page.....	27
7.8	Guide ( <i>Tools page</i> ).....	28
7.9	Kick Counter ( <i>Tools page</i> ).....	29
7.10	Weight Tracker ( <i>Tools page</i> ) .....	30
7.11	My Reminder ( <i>Tools page</i> ) .....	33
7.12	Information Page.....	34
7.13	About App ( <i>Information page</i> ) .....	34
7.14	Contact Us ( <i>Information page</i> ) .....	35
7.15	Location of Specialists ( <i>Information page</i> ).....	38
7.16	Current Location ( <i>Information page</i> ).....	39

Figure 1 - Architecture of the proposed system .....	6
Figure 2 - Panel .....	7
Figure 3 - Navigation bar.....	7
Figure 4 - Listview autodividers .....	8
Figure 5 - Full width collapsible listview .....	8
Figure 6 - Swipe.....	9
Figure 7 - Swipe script.....	9
Figure 8 - Tap .....	9
Figure 9 - Weight Tracker.....	10
Figure 10 - Weight tracker listview .....	10
Figure 11 - Chart progress of recorded weight.....	10
Figure 12 - Weight tracker listview .....	11
Figure 13 - My Reminder.....	12
Figure 14 - My reminder listview .....	12
Figure 15 – Creating table for permanent offline storage .....	13
Figure 16 – Insert and execute statement for permanent offline storage .....	14
Figure 17 - Appreminder table permanent offline storage .....	14
Figure 18 - weightracker table permanent offline storage.....	15
Figure 19 - Contact us form .....	15
Figure 20 - Contact us form validation.....	16
Figure 21 - Contact us form email validation.....	16
Figure 22 - Splash screen (1) .....	17
Figure 23 - Splash screen (2) .....	17
Figure 24 - Geolocation.....	18
Figure 25 – Current Weather .....	19
Figure 26 – Current Weather .....	19
Figure 27 - Kick Counter .....	20
Figure 28 - Kick Counter Script(1) .....	20
Figure 29 - Kick Counter Script(2) .....	21
Figure 30 - Kick Counter Script(3) .....	21
Figure 31 - Homepage .....	22
Figure 32 - Weekly Information .....	22
Figure 33 - Weekly Information .....	22
Figure 34 - Weekly Information .....	23
Figure 35 - Pregnancy Timeline.....	23
Figure 36 - Pregnancy Timeline.....	23
Figure 37 - Pregnancy Timeline.....	24
Figure 38 - Pregnancy Timeline.....	24
Figure 39 - My Bump .....	25
Figure 40 - My Bump .....	25
Figure 41 - Ultra Sound Image Page.....	25
Figure 42 - Ultra Sound Image .....	26
Figure 43 – Symptoms Page .....	26
Figure 44 - Symptoms .....	26
Figure 45 – Symptoms.....	27
Figure 46 - Symptoms .....	27
Figure 47 – Tools Page .....	27
Figure 48 – Guide .....	28
Figure 49 – Guide .....	28
Figure 50 Guide .....	28
Figure 51 – Panel.....	29

Figure 52 - Kick Counter .....	29
Figure 53 - Kick Counter .....	29
Figure 54 - Kick Counter .....	30
Figure 55 - Kick Counter .....	30
Figure 56 - Weight Tracker .....	30
Figure 57 - Weight Tracker .....	31
Figure 58 - Weight Tracker .....	31
Figure 59 - Weight Chart .....	32
Figure 60 - Weight Chart .....	32
Figure 61 - Reminder .....	33
Figure 62 - Reminder .....	33
Figure 63 - Reminder .....	33
Figure 64 - Information Page .....	34
Figure 65 - About App .....	34
Figure 66 - Privacy Policy .....	34
Figure 67 - Contact Us .....	35
Figure 68 - Validation .....	35
Figure 69 - Validation .....	35
Figure 70 – Validation .....	36
Figure 71 - Validation .....	36
Figure 72 - Validation .....	36
Figure 73 – Validation .....	37
Figure 74 - Validation .....	37
Figure 75 - Validation .....	37
Figure 76 – Location (1) .....	38
Figure 77 – Location (2) .....	38
Figure 78 - Current Location .....	39

## 1. Introduction

Pregnancy apps are currently available across the world. Engaging with pregnancy apps appears to have become a routine part of the maternity experience for many expectant mothers. Pregnancy apps provide you the freedom and comfort of keeping track of everything related to your unborn child's health and well-being. A pregnancy app may help take the guesswork out of the most important questions and concerns, from medical appointments to fetal growth to tracking the expectant woman's weight. Kick counter, weight tracker, reminders, data storage capability and images to track the pregnancy stage. Personalized tools are used to assess nutrition and health which are among the Web-based and mobile elements to consider while looking at pregnancy apps. Women also like pregnancy apps and digital platforms that are multifunctional and interact with one another. The high uptake of apps by women can be attributed to a number of reasons.

## 2. Problems

A lack of health-care professional guidance for women seeking information is indeed a source of concern. Younger women with less pregnancy experience, as well as first-time mothers, experience numerous difficulties and are especially vulnerable to unreliable information sources, as they are often active information seekers but less likely to understand what to expect during pregnancy and childbirth. As a result of their information seeking, young first-time mothers showed higher anxiety and tension. Women's Web-based research can become a stressful exercise in which they struggle to know when to stop looking, where unnecessary concern is created and the overall result might be more questions or confusion than answers. First-time mothers have a limited knowledge of fetal development, pregnancy-related changes in their bodies, and pregnancy-related risks and diseases. They tend to forget important tasks, such as medical appointments, as a result of anxiety and stress. These findings highlight the importance of health professionals being well-versed in relevant information and resources.

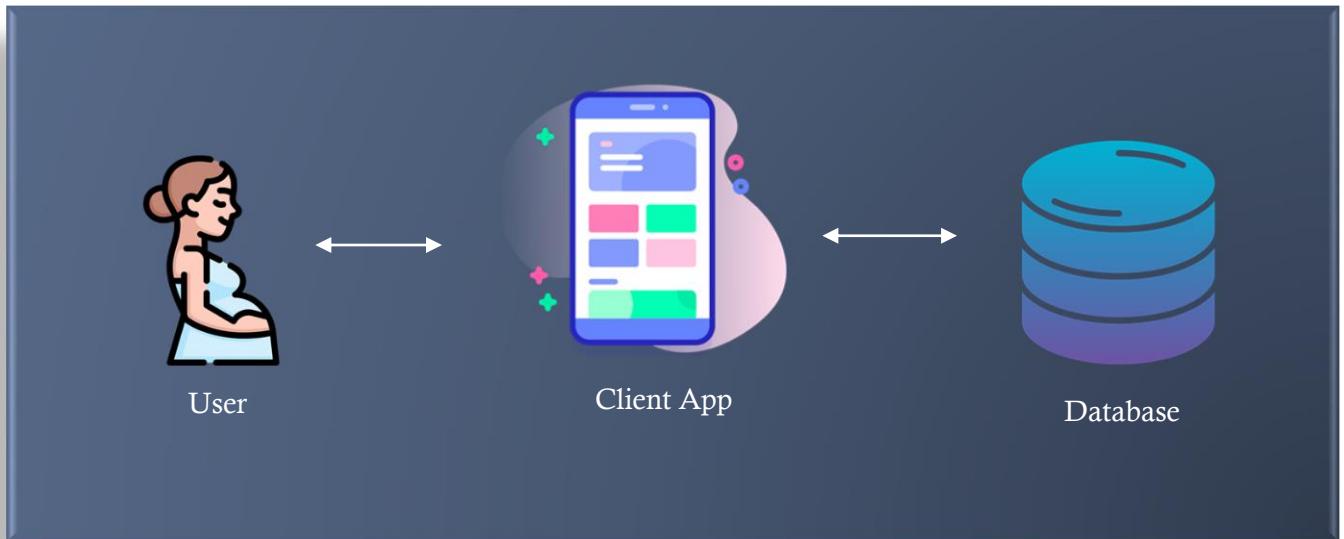
## 3. Scope

The pregnancy app will allow pregnant women to have access from medical checkups to fetal growth by tracking the pregnant woman's weight. The pregnancy app helps takes the guesswork out of the most important questions and concerns. To track the pregnancy stage, several functionalities have been implemented such as kick counter, weight tracker, reminders, data storage, and images. Web-based and mobile elements to consider while looking at pregnancy apps include personalized tools for assessing information related to nutrition, sport, rest and a better environment. Moreover, users can view location of several specialists and the current weather at their respective place from the app.

## 4. Use Case

- Users can easily navigate through the pregnancy app since the user interface is user-friendly.
- Once the user goes onto the homepage, she will see different options such as the Weekly Information, Pregnancy Timeline, My Bump and Ultrasound Images where she can navigate through to receive more information.
- Users can search for symptoms related to pregnancy and their solutions in the symptoms page.
- Users can go through different functionalities in the tools page.
- Users will get access to the guide page where she will receive information regarding health and nutrition.
- Users can record the kick of the baby per day in the kick counter page where actual date, start and finish time, and period is displayed.
- Users can keep track of their weight in the weight tracker page. The user should enter the actual weight, week of pregnancy and date.
- Users can view their weight progress which is displayed in the chart.
- Users can keep record of all the appointment in the reminder page. The user should enter the title, appointment, date and time.
- Users can get access to the information page where she can navigate through Contact Us page, Location Page and Current Weather page.
- Users can send query in the contact us page where their email will be verified.
- Users can get access to different specialists through the location page.
- Users can view the current weather at their respective place.

## 5. Architecture of the proposed system



*Figure 1 - Architecture of the proposed system*

Application architecture refers to a set of technologies and methods for creating the mobile app that are based on user demands. Programs operate on wireless devices like smartphones and tablets when the architecture of the app is built.

## 6. Solution

Information seeking is a key motivation for women using pregnancy apps. The pregnancy app allows users to instantly and easily obtain information mainly about fetal development, pregnancy-related changes in women's bodies, the types of food to consume, and the environment to stay during the pregnancy process. The app consists of different functionalities that allow pregnant women, particularly young first-time mothers, to keep track of their pregnancy progress as shown below:

➤ Functionalities that have been implemented:

- a. Use of jQuery Mobile UI features like panel, lists, navigation bar, and others

```
<div data-role="header" >
    <a href="#guide" class="ui-btn ui-btn-inline ui-icon-back ui-btn-icon-notext ui-corner-all" style="margin: 8px;"></a>
    <a href="#right-panel" data-icon="carat-l" data-iconpos="notext"
        class="ui-nodisc-icon-right ui-btn-inline" style="float: right; margin: 7px;"></a>
</div>
<!--Panel-->
<div data-role="panel" id="right-panel" data-display="push" data-theme="c"
style="border: 1px solid black; data-position="right" data-theme="b">

    <div style="text-align: center">
        
    </div>

    <a href="#NutFruits" data-rel="close" id="panelbtn" class="ui-btn">Fruits</a>
    <a href="#NutVeg" id="panelbtn" class="ui-btn">Vegetables</a>
    <a href="#NutVit" id="panelbtn" class="ui-btn">Vitamins</a>
    <a href="#NutWater" id="panelbtn" class="ui-btn">Water</a>

    <a href="#home" id="panelbtn" data-rel="close" class="ui-btn ui-corner-all ui-shadow ui-icon-home ui-btn-icon-right">Home</a>
    <a href="#" id="panelbtn" data-rel="close" class="ui-btn ui-corner-all ui-shadow ui-icon-delete ui-btn-icon-right">Close</a>
</div>
<!--End Panel-->
```

Figure 2 - Panel

Panels are a type of container control that can be placed on the UX Component. Panels are useful for grouping components and simplifying component layout.

```
<div data-role="navbar" class="nav">
    <ul>
        <li><a href="#home" class="ui-btn-active" data-icon="home">Home</a></li>
        <li><a href="#Symptoms" data-icon="grid">Symptoms</a></li>
        <li><a href="#tools" data-icon="alert">Tools</a></li>
        <li><a href="#infos" data-icon="info">Information</a></li>
    </ul>
</div>
```

Figure 3 - Navigation bar

Navigation bar is a section of a graphical user interface that helps users to easily access information.

```

<div class="symptom-content">
  <ul data-role="listview" id="SymptomList" data-autodividers="true" data-filter="true" data-inset="true">
    <li>
      <div data-role="collapsible" class="collaps">
        <h3>Abdominal Itching</h3>
        <b>What's that?</b>
        <p>Oh your belly seems to be bitten by a swarm of bees, it itches so much! You are not alone: skin irritation on a growing belly is a very common problem beginning in the fifth month of pregnancy. When skin expands, it becomes increasingly moisture-deprived and dry skin often feels itchy and uncomfortable.</p>
        <br/>
        <b>What to do now?</b>
        <p>You don't need to scratch all the time! Use these tips.</p>
        <i><p>Don't scratch all the time.</p>
          <p>- Try anti-itch cream.</p>
          <p>- Buy vitamin E oil.</p>
          <p>- Take a warm bath.</p>
          <p>- Use a humidifier.</p>
          <p>- Let it go.</p></i>
      </div>
    </li>
    <li>...
    </li>
    <li>...
    </li>
  </ul>

```

Figure 4 - Listview autodividers

The listview autodivider displays a collection of items and provide several ways to arrange and display items that are more flexible.

```

<div data-role="collapsibleset" data-inset="false">
  <div data-role="collapsible" id="GuideList">
    <h2>Nutrition<span class="ui-li-count">4</span></h2>
    <ul data-role="listview">
      <li class="collapsGuide"><a href="#NutFruits">Fruits</a></li>
      <li class="collapsGuide"><a href="#NutVeg">Vegetables</a></li>
      <li class="collapsGuide"><a href="#NutVit">Vitamins</a></li>
      <li class="collapsGuide"><a href="#NutWater">Water</a></li>
    </ul>
  </div>
  <div data-role="collapsible" id="GuideList">
    <h2>Sport<span class="ui-li-count">3</span></h2>
    <ul data-role="listview">
      <li class="collapsGuide"><a href="#SportYoga">Yoga</a></li>
      <li class="collapsGuide"><a href="#SportSquats">Squats</a></li>
      <li class="collapsGuide"><a href="#SportCycling">Cycling</a></li>
    </ul>
  </div>
  <div data-role="collapsible" id="GuideList">
    <h2>Rest<span class="ui-li-count">2</span></h2>
    <ul data-role="listview">
      <li class="collapsGuide"><a href="#RestBath">Bath</a></li>
      <li class="collapsGuide"><a href="#RestSleep">Sleep</a></li>
    </ul>
  </div>
  <div data-role="collapsible" id="GuideList">
    <h2>Influence of the environment <span class="ui-li-count">3</span></h2>
    <ul data-role="listview">
      <li class="collapsGuide"><a href="#EnvironmentPets">Domestic pets</a></li>
      <li class="collapsGuide"><a href="#EnvironmentHome">Home environment</a></li>
      <li class="collapsGuide"><a href="#EnvironmentPAir">Polluted air</a></li>
    </ul>
  </div>
</div>

```

Figure 5 - Full width collapsible listview

Full width collapsible listview shows an indicator beside each item to display the item's current state. (The states are usually one of expanded group, collapsed group, child, or last child)

b. Specific mobile related events like swipe, rotate, tap, and others

```
<div class="swiper">
  <!-- Additional required wrapper -->
  <div class="swiper-wrapper">
    <!-- Slides -->
    <div class="swiper-slide"><h5 style="text-align: center;">Week 1</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 2</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 3</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 4</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 5</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 6</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 7</h5><hr/><br/>
    <div class="swiper-slide"><h5 style="text-align: center;">Week 8</h5><hr/><br/>
```

Figure 6 - Swipe

Figure 6 shows a swipe event which indicates that the user performed a swipe gesture such as dragging a finger in one direction on touch screen and the information has been hardcoded.

The swipe event is triggered when the user swipe over an element horizontally.

The swipe functionality has been implemented to represent some pictures as shown in figure 6 and a loop has been used so as when swiping even after the last image, it resumes with the first image.

```
<!-- Script Slider main container -->
<script>
  const swiper = new Swiper('.swiper', {
    // Optional parameters
    direction: 'horizontal',
    loop: true,

    // If we need pagination
    pagination: {
      el: '.swiper-pagination',
    },

    // Navigation arrows
    navigation: {
      nextEl: '.swiper-button-next',
      prevEl: '.swiper-button-prev',
    },

    // And if we need scrollbar
    scrollbar: {
      el: '.swiper-scrollbar',
    },
  });
</script>
```

Figure 7 - Swipe script

```
<!--Terms-->
<a href="#popupCloseRightT" data-rel="popup" data-position-to="window" data-transition="flip" class="button">
  <div class="item">
    <h4>Terms of Use</h4>
  </div>
</a>
<div data-role="popup" id="popupCloseRightT" class="ui-content" style="max-width:100%; background-color: #aliceblue;">
  <a href="#" data-rel="back"
    class="ui-btn ui-corner-all ui-shadow ui-btn-a ui-icon-delete ui-btn-icon-notext ui-btn-right">Close</a>
  <b>1. GENERAL TERMS</b>
  <p style="text-align: justify;">1.1. The mobile application "Pregnancy" was made by and is the property of the company TimskiyApp ("We", "The Company"). The current User Agreement ("Agreement") regulates your usage ("User", "You") of the "Pregnancy" application, as well as other goods that we offer.</p>
  <p style="text-align: justify;">1.2. The "Pregnancy" mobile application (referred to as "The Application") is a software application store on the conditionality that the Agreement is fully adhered to. The Application is an informative application, developed for mobile devices operating the Android and Apple iOS operating systems. The right to the intellectual property of The Application is owned by The Company.</p><br/>
  <b>2. TERMS OF USE</b>
  <p style="text-align: justify;">2.1. The User is obliged to duly follow the conditions of our current Agreement.</p>
  <p style="text-align: justify;">2.2. The User commits to not use The Application for any purpose other than for the purpose of non-<br/><b>3. EXCLUSIVE RIGHTS TO THE APPLICATION</b>
  <p style="text-align: justify;">3.1. The use of the contents of The Application by the User for personal, non-commercial use is allowed, provided that all signs of copyright protection, related rights, trademarks, and other notices of authorship, and name of authorship are preserved.</p>
</div>
```

Figure 8 - Tap

Figure 8 shows a tab where you lightly touch a spot on the screen so as to display information.

### c. Weight Tracker

```
<div class="weightTracker-form">
  <h2 style="font-family: sans-serif; letter-spacing: 2px; text-align: center;">Weight Tracker Form</h2><hr/><br/>

  <form method="post" onSubmit="return weighttrack()">
    <label style="font-weight: bold; font-family: sans-serif;" for="weight">Weight:</label>
    <input name="weight" type="number" id="weight" placeholder="Enter weight in KG...">

    <label style="font-weight: bold; font-family: sans-serif;" for="w_week">Week:</label>
    <input name="w_week" type="text" id="w_week" placeholder="Enter current week (e.g week1)">

    <label style="font-weight: bold; font-family: sans-serif;" for="w_date">Date:</label>
    <input name="w_date" type="date" id="w_date">
    <br/>
    <input style="background: #rgb(83,130,154); background: linear-gradient(90deg, #rgb(83,130,154,1) 0%, #rgba(100,219,224,1) 50%, #rgba(83,130,154,1) 100%);" type="submit" value="CONFIRM"/>
  </form>
</div>

<div class="chart" style="text-align: end; margin-top: 10px;">
  <a href="#chart" style="margin: 12px; color: #5d903a; font-weight: 12px;">Chart Progress</a>
</div>
```

Figure 9 - Weight Tracker

Figure 9 shows a weight tracker form where it allows pregnant women to track and stay within a healthy weight range through their pregnancy from conception to delivery. An **href** has been used to redirect the user to the progression chart page.

```
<h2 style="font-family: sans-serif; letter-spacing: 2px; text-align: center; color: #rgb(43, 94, 14);">Weight Tracker</h2>
<hr/><br/>
<ul data-role="listview" id="resultListWeightTrack"></ul>
```

Figure 10 - Weight tracker listview

Figure 10 shows the implemented code where it provides a detailed assessment of their inserted weight which is displayed in a listview and helps log and visualize the progress until the baby arrives which is then shown in a line graph progression chart. This information is stored in a permanent offline database.

An array has been passed to represent the inserted weight (*const w*) and date (*const w\_d*) as shown in figure 12 below.

In figure 11, to display the inserted information in a chart, **labels: w\_d** represents the date which will be in the x-axis and **data: w** represents the weight which will be in the y-axis. These are the code to display the inserted information a line graph chart.

```
//Chart
var canvas = document.getElementById('myChart');
var data = {
  labels: w_d,
  type: "line",
  datasets: [
    {
      label: "Weight",
      backgroundColor: "rgba(255,99,132,0.2)",
      borderColor: "rgba(255,99,132,1)",
      borderWidth: 1,
      hoverBackgroundColor: "rgba(255,99,132,0.4)",
      hoverBorderColor: "rgba(255,99,132,1)",
      data: w,
    }
  ];
};

var myBarChart = Chart.Line(canvas,{
  data:data,
});
//End Chart
```

Figure 11 - Chart progress of recorded weight

```

function successQueryDB1(tx){
    //alert("Insert successful");
    tx.executeSql('SELECT * FROM weighttracker', [], renderListCB, errorCB);
    tx.executeSql('SELECT * FROM Appreminder', [], renderListCB_, errorCB);
}

function renderListCB(tx, results){
    var htmlString = ' ';
    var len = results.rows.length;
    const w = [];

    for(var i=0; i<len;i++){
        htmlString+=<li>+<b>'Weight: '+</b>+results.rows.item(i).weight+<br/>+<b>'Week: '+</b>+results.rows.item(i).w_week+';
        //htmlString+= results.rows.item(i).weight;
        //htmlString+=",";
        w.push(results.rows.item(i).weight);
    }

    //alert(htmlString);
    //alert(w);
    var len2 = results.rows.length;
    const w_d = [];

    for(var i=0; i<len2;i++){
        w_d.push(results.rows.item(i).w_date);
    }
    //alert(w_d);

    $('#resultListWeightTrack').html(htmlString);
    //$('#resultListWeightTrack').listview('refresh');
}

```

Figure 12 - Weight tracker listview

A *function test()* as shown in *figure 15* below, has been used so that when the page load, it will perform the *successQueryDB1* and in the *function successQueryDB1(tx)* as shown in figure 12, the renderlist will be executed so that the inserted information of weight tracker can be displayed in a listview and also that in the chart as mentioned above.

For the listview, a loop has been used to repeat the specific block until the last number and the line *htmlString+=* has been used to display the result. For the chart *w.push(results.rows.item(i).weight)* and *w\_d.push(results.rows.item(i).w\_date)* have been used inside a for loop to insert elements to the end of the array.

After the looping process the *\$('#resultListWeightTrack').html(htmlString)* is used to displayed the information where it matches the *#resultListWeightTrack* id.

#### d. My Reminder

```
<div class="reminder-form">
    <h2 style="text-align: center; letter-spacing: 2px; font-family: sans-serif;">Reminder Form</h2><br/>
    <form method="post" onSubmit="return insertReminder()">
        <label style="font-weight: bold; font-family: sans-serif;" for="r_title">Title:</label>
        <input name="r_title" type="text" id="r_title" placeholder="Enter Title">

        <label style="font-weight: bold; font-family: sans-serif;" for="r_appointment">Appointment:</label>
        <input name="r_appointment" type="text" id="r_appointment" placeholder="Enter Appointment">

        <label style="font-weight: bold; font-family: sans-serif;" for="r_date">Date:</label>
        <input name="r_date" type="date" id="r_date">

        <label style="font-weight: bold; font-family: sans-serif;" for="r_time">Time:</label>
        <input name="r_time" type="time" id="r_time">
        <br/>
        <input style="background: linear-gradient(90deg, #rgb(83,130,154), #rgba(83,130,154,1) 0%, #rgba(100,219,224,1) 50%, #rgba(83,130,154,1) 100%); type="submit" value="CONFIRM"/>
    </form>
</div>
<br/><br/><br/>
<!--display reminder-->
<h2 style="font-family: sans-serif; letter-spacing: 2px; text-align: center; color: #rgb(43, 94, 14);">My Reminder</h2>
<hr/><br/>
<ul data-role="listview" id="DailyReminder"></ul>
<!--End display reminder-->
<br/>
```

Figure 13 - My Reminder

The reminder form in figure 13, serves as purpose of keeping mothers focused and on track. It saves the important events of user that they have entered. The information is then displayed in a listview by the id **DailyReminder** and stored in a permanent offline database.

```
function renderListCB(tx, results){
    var htmlString = '';
    var len = results.rows.length;

    for(var j=0; j<len;j++){
        htmlString+="- Title: "+results.rows.item(j).r_title+"  
Appointment: "+results.rows.item(j).r_appointment+"  
"
                    +"Date: "+results.rows.item(j).r_date+"  
Time: "+results.rows.item(j).r_time+"


---

";
    }

    $('#DailyReminder').html(htmlString);
    //$('#resultList1').listview('refresh');
}
```

Figure 14 - My reminder listview

A function **test()** as shown in figure 15 below, has been used so that when the page load, it will perform the **successQueryDB1** and in the function **successQueryDB1(tx)** as shown in figure 12, the renderlist will be executed so that the inserted information of reminders can be displayed in a listview.

For the listview, a loop has been used to repeat the specific block until the last number and the line **htmlString+=** has been used to display the result. After the looping process the **\$('#DailyReminder').html(htmlString)** is used to displayed the information where it matches the **# DailyReminder** id.

## e. Permanent Offline storage features

```
/*Weight Tracker + Reminder*/
document.addEventListener("deviceready", onDeviceReady, false);
var db;
function onDeviceReady(){
    //alert("device ready");

    db=window.openDatabase("weighttrackerDB", "1.0", "Simple Demo", 2*1024*1024);
    db.transaction(createDB, errorCB, successCB);
    test();
}

function test() {
    db.transaction(successQueryDB1, errorCB);
    return false;
}

function createDB(tx){
    tx.executeSql('CREATE TABLE IF NOT EXISTS weighttracker(id INT, weight INT, w_week VARCHAR, w_date DATE)');
    tx.executeSql('CREATE TABLE IF NOT EXISTS Appreminder(id INT, r_title TEXT, r_appointment TEXT, r_date DATE, r_time TIME)');
}

function errorCB(err){
    alert("SQL Error: "+err.code);
}

function successCB(){
    //alert("Database & Table Created");
}

function weightrack(){
    db.transaction(insertDB, errorCB);
    return false;
}

function insertReminder(){
    db.transaction(insertDB_, errorCB);
    return false;
}
```

Figure 15 – Creating table for permanent offline storage

Offline storage refers to any storage medium that must be physically inserted into a system every time a user wants to access or edit data. In figure 15, it shows a database **weighttrackerDB** has been created along with two tables called **weighttracker** and **Appreminder**.

```

function insertDB(tx){
    var weight_c=$("[name='weight']").val();
    var w_week_c=$("[name='w_week']").val();
    var w_date_c=$("[name='w_date']").val();
    var sql='INSERT INTO weighttracker (weight, w_week, w_date) VALUES (?,?,?)';
    tx.executeSql(sql,[weight_c,w_week_c,w_date_c], successQueryDB, errorCB);
}

function insertDB_(tx){
    var r_title_c=$("[name='r_title']").val();
    var r_appointment_c=$("[name='r_appointment']").val();
    var r_date_c=$("[name='r_date']").val();
    var r_time_c=$("[name='r_time']").val();
    var sql='INSERT INTO Appreminder (r_title, r_appointment, r_date, r_time) VALUES (?,?,?,?,?)';
    tx.executeSql(sql,[r_title_c,r_appointment_c,r_date_c,r_time_c], successQueryDB_, errorCB);
}

function successQueryDB(tx){
    alert("Insert Successful");
    tx.executeSql('SELECT * FROM weighttracker', [], renderListCB, errorCB);
    weight.value = '';
    w_week.value = '';
    w_date.value = '';
}

function successQueryDB_(tx){
    alert("Insert Successful");
    tx.executeSql('SELECT * FROM Appreminder', [], renderListCB_, errorCB);
    r_title.value = '';
    r_appointment.value = '';
    r_date.value = '';
    r_time.value = '';
}

```

Figure 16 – Insert and execute statement for permanent offline storage

Figure 16 shows the insert and execute statement for the two tables as mentioned above. After an insert, an alert *insert successful* will appear and also that all the textboxes will be cleared.

	rowid	id	r_title	r_appointment	r_date	r_time
Application	1		undefined	undefined	undefined	undefined
	2		undefined	undefined	undefined	undefined
Manifest	3		Headache	Doctor	2022-01-22	01:59
Service Workers	4		Fever	Hospital	2022-02-05	03:05
Storage	5		Back Pain	Consultant	2022-02-01	05:12
	6		Belly Pain	Nurse	2022-02-04	06:50
Local Storage						
Session Storage						
IndexedDB						
Web SQL						
weighttrackerDB						
Appreminder						
reminder						
weightracker						
Cookies						
Trust Tokens						

Figure 17 - Appreminder table permanent offline storage

Figure 17 shows the inserted information in the table name **Appreminder** under **weightrackerDB** database.

The screenshot shows the Chrome DevTools Application tab. On the left, there's a tree view of application components: Manifest, Service Workers, Storage, Local Storage, Session Storage, IndexedDB, Web SQL, weighttrackerDB (which is expanded to show Appreminder, reminder, and weighttracker), Cookies, and Trust Tokens. On the right, a table named 'weighttracker' is displayed with the following data:

rowid	id	weight	w_week	w_date
1		50	week1	2022-01-18
2		51	week1	2022-01-19
3		52	week1	2022-01-20
4		53	week2	2022-01-21
5		54	week2	2022-01-22
6				
7				
8				
9		55	week2	2022-01-23
10		56	week2	2022-01-24
11		57	week2	2022-01-25
12				
13		58	week1	2022-01-26

Figure 18 - weighttracker table permanent offline storage

Figure 18 shows the inserted information in the table name **weighttracker** under **weighttrackerDB** database.

#### f. Web service-related functionality (*Contact Us*)

```
<div class="ContactWrapper" >
    <h2 style="text-align: center; font-family: sans-serif; letter-spacing: 3px;">CONTACT US</h2><hr/><br/>
    <form id="myform" >
        <div class="input_field">
            <input type="text" placeholder="Full Name" id="name">
        </div>
        <div class="input_field">
            <input type="text" placeholder="Phone" id="phone">
        </div>
        <div class="input_field">
            <input type="text" placeholder="subject" id="subject">
        </div>
        <div class="input_field">
            <textarea id="message" placeholder="Message"></textarea>
        </div>
        <div class="input_field">
            <input type="email" placeholder="Email" id="email">
        </div><br/>
        <hr/>
        <div class="btn">
            <a href="#" class="ui-btn ui-corner-all" id="submitbtn" onclick="contactvalidation()">Submit</a>
            <input type="reset" value="Reset">
        </div>
    </form>
</div>
```

Figure 19 - Contact us form

As shown in figure 19, reset button has been used to clear the form and a submit button has been used to validate the form. The contact us page exists to serve the user with the purpose on how they can get in touch regarding any query.

```

function contactvalidation() {
    if (document.getElementById('name').value == "" || document.getElementById('name').value == undefined)
    {
        alert ("Please Enter Valid Name");
        return false;
    }

    if(document.getElementById('phone').value == "" || document.getElementById('phone').value == undefined)
    {
        alert ("Please Enter Valid Phone Number");
        return false;
    }

    if(document.getElementById('subject').value == "" || document.getElementById('subject').value == undefined)
    {
        alert ("Please Enter Correct Subject");
        return false;
    }

    if(document.getElementById('message').value == "" || document.getElementById('message').value == undefined)
    {
        alert ("Please Enter A Message");
        return false;
    }

    if(document.getElementById('email').value == "" || document.getElementById('email').value == undefined)
    {
        alert ("Please Enter your Email");
        return false;
    }
}

```

Figure 20 - Contact us form validation

Figure 20 shows the validation for the contact us form. Alerts will appear if the textboxes are null or undefined.

```

if (/[^@]+@[^@]+\.[^@]+/.test(document.getElementById('email').value))
{
    var emailAddress = document.getElementById('email').value
    var apiKey = "7fd2e60c-e3b4-4693-9d4c-2699eefcba1a"

    $.ajax("https://isitarealemail.com/api/email/validate?email=" +
    emailAddress,
    {
        crossDomain: true,
        headers: {
            Authorization: "Bearer " + apiKey
        }
    })
    .then(function responseHandler(data) {
        if (data.status === 'valid') {
            console.log("the email is valid and the mail box is active")
            //alert("valid");
            alert ("Form Submitted Successfully!");

            $(name).val(" ");
            $(phone).val(" ");
            $(subject).val(" ");
            $(message).val(" ");
            $(email).val(" ");

        } else {
            console.log("the email is incorrect or unable to be tested.")
            alert("Please Enter Valid Email");
        }
    });
    return false;
} else {
    alert ("Please Enter a Valid Email");
    return false;
}
}

```

Figure 21 - Contact us form email validation

Figure 21 shows the validation for the email. User must enter an existing email so that they can get in touch. The first if statement has been used to check if the email is in a valid format else an alert will appear. An API has been utilized where the **var apiKey** has been used to stored my *api key* and a Real

Email API Validation, the url inside the ajax, has been used to check whether the inserted email is an existing email or not.

The second if statement has been used to execute the form if the email is valid and also clear all the textboxes else an alert will appear.

#### g. Splash Screen

```
<script type="text" id="splash-screen">
    <div class="splash card">
        <br/>
        <p class="lead" style="text-align: center">
            <h2 style="font-family: sans-serif; text-align: center">Pregnancy App...</h2><br/>
        </p>
        <div class="progress">
            <div class="mybar" role="bar"></div>
        </div>
    </div>
</script>
```

Figure 22 - Splash screen (1)

Figure 22 shows the implemented code for the splash screen. The **role="bar"** has been used so that when a user enters the app the loading bar will appear.

Figure 23 shows that the splash screen has been initialized.

```
$(function(){
    NProgress.configure({
        template: $('#splash-screen').html()
    });
    NProgress.start();
});
$(window).load(function(){
    NProgress.done();
})
```

Figure 23 - Splash screen (2)

## h. Geolocation services using the OSM Map Service (*Location of Specialists*)

```
<!--contact us form-->
<div id="mapid" style="width: 100%; height: 470px;"></div>
<br/><br/>
<h4>Location of specialists in Mauritius</h4>
<p style="text-align: justify;">It's important to find a primary care doctor located close to your home or work. When the main...
</p><br/>
<h4>Benefits of a Convenient Location</h4>
<p style="text-align: justify;">Primary care doctors, such as family practice physicians or internists, are a go-to source when ...
</p><br/>
<h4>Less Time in Traffic</h4>
<p style="text-align: justify;">Finding a doctor close to your home makes it easy to fit in a visit on the way to or from work, and ...
</p><br/>
<h4>Address Health Concerns Promptly</h4>
<p style="text-align: justify;">If seeing the doctor involves a drive across town or a commute, you're more likely to put off health...
</p><br/>
<h4>Follow-Up is Easy</h4>
<p style="text-align: justify;">If your doctor recommends follow-up care, it's much more convenient if the office is close by. You'll...
</p>
<script>
    var mymap = L.map('mapid').setView([-20.134557, 57.500880], 10);
    L.tileLayer('https://s.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        maxZoom: 19,
        attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
    }).addTo(mymap);

    var marker = L.marker([-20.134557, 57.500880]).addTo(mymap);
    marker.bindPopup("<b>Dr Mukesh Sooknundun</b>").closePopup();
    var circle = L.circle([-20.134557, 57.500880], {
        color: 'red',
        fillColor: '#f03',
        fillOpacity: 0.5,
        radius: 100
    }).addTo(mymap);

    var marker1 = L.marker([-20.312329, 57.518846]).addTo(mymap);
    marker1.bindPopup("<b>Dr .Ashok Anand Manraj</b>").closePopup();
    var circle = L.circle([-20.312329, 57.518846], { ...
```

Figure 24 - Geolocation

Figure 24 shows the geolocation which have been hardcoded. **marker** has been used to display specialists in Mauritius along with their respective names by using **bindPopup**.

➤ Functionalities that have not been able implemented:

a. Current Weather

```
.currentWeather{
    width: 100%;
    animation: mymove 5s infinite;
    margin: auto;
    text-align: center;
}
@keyframes mymove {
    100% {background-position: center;}
}
.weatherbg{
    width: 100%;
    background-image: url(images/weather.gif);
    color: cornsilk;
    animation: mymove 5s infinite;
    border-radius: 1%;
    padding: 1em;
    text-align: center;
}
@keyframes mymove {
    100% {background-position: center;}
}
#weatherpage{
    border-radius: 1%;
}
```

Figure 25 – Current Weather

Figure 25 shows the style of the functionality Current Weather.

```
4241 <div data-role="popup" id="weatherpage">
4242     <script>
4243         if (navigator.geolocation) {
4244             navigator.geolocation.getCurrentPosition(function(position) {
4245                 var latitude = position.coords.latitude
4246                 var longitude = position.coords.longitude
4247                 console.log(latitude, longitude);
4248
4249                 $.getJSON("http://api.openweathermap.org/data/2.5/weather?lat=" + latitude + "&lon=" +
4250                     longitude + "&units=metric&appid=3731a5e928edfa6a4ab6c334e7a3957c",
4251                     function(data) {
4252                         console.log(data);
4253                         var text = "${data.name}"
4254
4255                         $("#weather").html(text);
4256                     });
4257
4258             });
4259         }
4260     </script>
4261 </div>
4262 <a href="#currentWeather" class="button" data-rel="popup" data-position-to="window" data-transition="slideup">
4263     <div class="item">
4264         <br/>
4265         <h4>Current Weather</h4>
4266     </div>
4267 </a>
4268 <div data-role="popup" id="currentWeather">
4269     <div class="weatherbg">
4270         <div class="app">
4271             <span>Temperature:</span>
4272             <h1 id="temp-main">0</h1><br/>
4273             <span>Condition:</span>
4274             <h3 id="condition">unknown</h3><br/>
4275             <span>Humidity:</span>
4276             <h2 id="humidity">unknown</h2>
4277     </div>
```

Figure 26 – Current Weather

The getCurrentPosition() has been used to get the use current position as shown in figure 26. An api has been utilize to get the weather location based on its coordinates and is sent to retrieve data. The data is in the form of a json file, which must be parsed. The items are processed and saved in a variable before being concatenated to an html id.

## b. Kick Counter

The screenshot shows a web application interface. At the top, there is a 'count-container' div containing a 'count' h1 element with the value '0'. Below it is a 'btms' div containing two buttons: one labeled 'reset' and another labeled 'plus'. Below this is a table with a header row containing columns for Date, Start Time, Finish Time, Period/s, and Quantity. The table body is currently empty.

```
<main>
  <div class="count-container">
    <h1 class="count" id="count">0</h1>
    <div class="btms">
      <button id="reset" onclick="reset()">reset</button>
      <button id="plus" onclick="plus()">+</button>
    </div>
  </div>
</main>
<br>
<table data-role="table" class="ui-responsive">
  <thead>
    <tr>
      <th>Date</th>
      <th>Start Time</th>
      <th>Finish Time</th>
      <th>Period/s</th>
      <th>Quantity</th>
    </tr>
  </thead>
  <tbody></tbody>
</table>
```

Figure 27 - Kick Counter

Figure 27 shows a kick counter that can help the mother learn about the baby's movement patterns and recognize if there is a change and also to keep track of the baby's movement every day.

The screenshot shows the script code for the 'plus()' function. The code initializes a 'table' array, sets up a date string, creates a new row object, and checks if a previous row exists. It then updates or creates a new row in the table based on the current date and time.

```
<script>
  // Initialise table array to hold row values
  var table = [];

  // Button plus handler
  function plus() {
    // current date again in plus() method in case date changes
    var now = new Date();

    // Set date part (ex. 2022-1-12)
    var date = now.getFullYear() + '-' + ((now.getMonth() + 1) < 10 ? '0' + (now.getMonth() + 1) : (now.getMonth() + 1)) + '-' +
    (now.getDate() < 10 ? '0' + now.getDate() : now.getDate());

    // Initialise a row object
    var row = null;

    // Try to get the last row by checking if
    // the table is not empty
    if (table.length > 0) {

      // Get the last row
      var lastIndex = table.length - 1;
      row = table[lastIndex];
    }

    // Check if a new row should be added or if the last row should be updated
    // To do so, we
    // - check if the last row was able to be fetched (!row)
    // - or if the last row's date is not equal to the current date (the date has changed)
    if (!row || row.date !== date) {
      // Set start time using current date
      var startTime = (now.getHours() < 10 ? '0' + now.getHours() : now.getHours()) + ":" + (now.getMinutes() < 10 ? '0' +
      now.getMinutes() : now.getMinutes());
    }

    // Create a new row
    row = {
```

Figure 28 - Kick Counter Script(1)

```

        row = {
            date: date,
            startTime: startTime,
            finishTime: startTime,
            period: 0,
            quantity: 1,
            startDate: now
        }

        // Add newly created row to the table
        table.push(row);

        // Add row in html
        $('tbody').append('<tr><td class="date">' + row.date + '</td><td class="start-time">' + row.startTime +
            '</td><td class="finish-time">' + row.finishTime +
            '</td><td class="period">' + row.period + '</td><td class="quantity">' + row.quantity + '</td></tr>');

    } else {
        // Set finish time
        var finishTime = (now.getHours() < 10 ? '0' + now.getHours(): now.getHours()) + ":" + (now.getMinutes() < 10 ? '0' + now.getMinutes(): now.getMinutes());

        // Calculate period (divide by 1000 to get period in seconds)
        var period = Math.round((now - row.startDate) / 1000);

        row.finishTime = finishTime;
        row.period = period;
        row.quantity = row.quantity + 1;

        var lastIndex = table.length - 1;
        $('tbody:last-child td.finish-time').html(row.finishTime);
        $('tbody:last-child td.period').html(row.period);
        $('tbody:last-child td.quantity').html(row.quantity);
    }

    document.getElementById("count").innerText = row.quantity;

```

Figure 29 - Kick Counter Script(2)

The kick counter has been implemented using a table array to hold row values. The date, start time, finish time, period, and quantity display in the table row as soon the user click on the plus button handler. A newly created row is added to the table. The table initializes a row object using the conditional loop (if/else statement). A validation has been added to the date variable. In case the date changes from 2022-02-01 to 2022-02-02, a new row will be added to the table.

```

        $('tbody:last-child td.finish-time').html(row.finishTime);
        $('tbody:last-child td.period').html(row.period);
        $('tbody:last-child td.quantity').html(row.quantity);
    }

    document.getElementById("count").innerText = row.quantity;
}

function reset() {
    // Try to get the last row by checking if
    // the table is not empty
    if (table.length > 0) {
        // Get the last row
        var lastIndex = table.length - 1;
        var row = table[lastIndex];
        row.finishTime = 0;
        row.period = 0;
        row.quantity = 0;
        $('tbody:last-child td.finish-time').html(row.finishTime);
        $('tbody:last-child td.quantity').html(row.quantity);
        $('tbody:last-child td.period').html(row.period);
    }

    document.getElementById("count").innerText = 0;
}

```

Figure 30 - Kick Counter Script(3)

In case the user presses the plus button before the kick, she can use the reset button to restart the count. The time, date, and quantity will all be reset. The reset function tries to get the last row by determining whether or not the table is empty.

**Difficulties faced:** we are not being able to store the kick counter in a database.

## 7. Testing

Testing has been done on an android device Samsung Galaxy A51

### 7.1 Homepage



Figure 31 - Homepage

Figure 31 shows the *homepage* of the pregnancy app which consists of different features.

### 7.2 Weekly Information (Homepage)

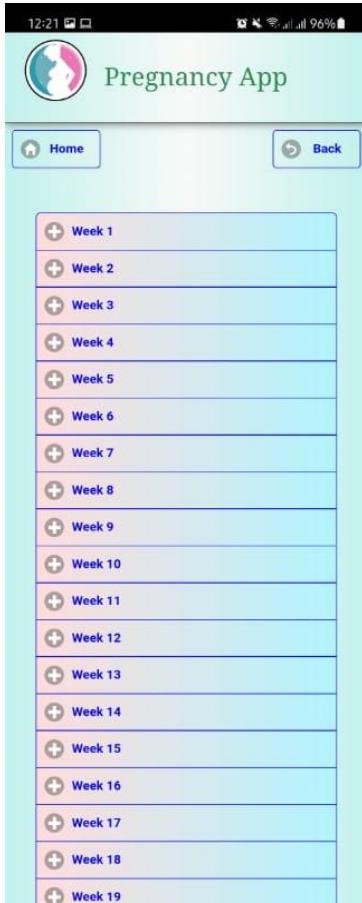


Figure 32 - Weekly Information

Figure 32 shows the *weekly information* where pregnant woman can easily access to this information.

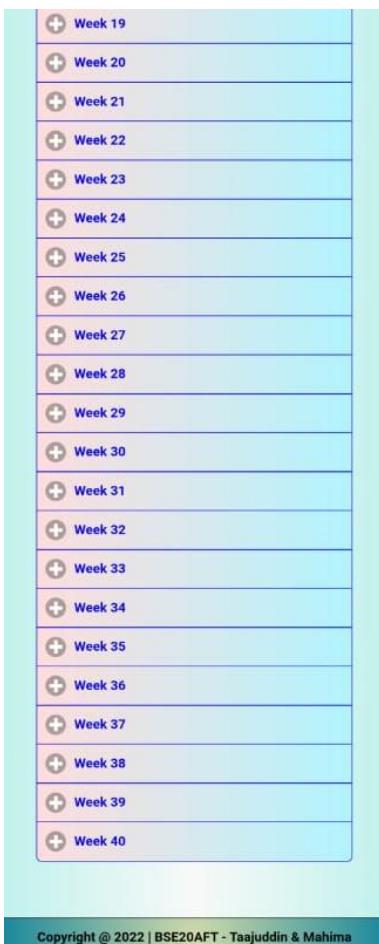


Figure 33 - Weekly Information

Figure 33 shows the *weekly information* of all weeks during the pregnancy process.

**Week 1**

**Baby**  
The body is continuing to prepare itself for pregnancy. The follicle resembles a bubble, filled with liquid containing a maturing oocyte. The dominant follicle is recognizable, as it has a bigger size than the others, and is closer to the surface of the ovaries. During this period the hormones start to secrete - oestrogen, which is necessary for the strengthening and thickening of the lining; and progesterone, to supply the fertilised oocyte with the proper nutrients after its implantation. After ovulation, the dominant follicle goes through all the stages of maturation and the others resorb (undergo atresia). The oocyte is viable for approximately 24 hours after ovulation - this time being the most important factor for conception to be successful. During ovulation, this particular oocyte moves along the fallopian tubes towards the one in a million spermatozoa which made its way through the egg's outer layer. This is why intercourse should happen during this period.

**Mum**  
You might have mild discomfort in the lower abdomen, and increased vaginal discharge during ovulation. Your body temperature might be slightly elevated; and nausea and fatigue may occur due to hormonal changes in the body and progesterone secretion. Increased attention should be given to intimacy with your partner, due to increased sexual drive.

**Helpful Tips**  
Try to predict when ovulation begins, as this period increases your chances of getting pregnant. For conception to be successful, you need to avoid all methods of contraception, and you should give extra attention to intimate relations with your partner. The research shows that the best way to get pregnant is by having intercourse frequently - especially every other day. The best moment for conception is three days before or on the day of ovulation. Keep looking after your own health, and spend more time outdoors.

**Week 2**

**Week 3**

Figure 34 - Weekly Information

Figure 34 shows all the related information based on each week which includes the baby, mother and there are also some helpful tips for the pregnant woman.

## 7.3 Pregnancy Timeline (Homepage)

**Pregnancy Stages:**

- I Trimester
- II Trimester
- III Trimester
- Birth

**Summary**

- Pregnancy is counted as 40 weeks, starting from the first day of the mother's last menstrual period. Your estimated date of birth is only to give you a guide. Babies come when they are ready and you need to be patient.
- The gender and inherited characteristics of the baby are decided at the moment of conception.

The unborn baby spends around 38 weeks in the womb, but the average length of pregnancy (gestation) is counted as 40 weeks. This is because pregnancy is counted from the first day of the woman's last period, not the date of conception, which generally occurs two weeks later.

Pregnancy is divided into three trimesters:
 

- First trimester** – conception to 12 weeks
- Second trimester** – 12 to 24 weeks
- Third trimester** – 24 to 40 weeks.

**Conception**  
The moment of conception is when the woman's ovum (egg) is fertilised by the man's sperm. The gender and inherited characteristics are decided in that instant.

**Week 1**  
This first week is actually your menstrual period. Because your expected birth date (EDD or EDB) is calculated from the first day of your last period, this week counts as part of your 40-week pregnancy, even though your baby hasn't been conceived yet.

**Week 2**  
Fertilisation of your egg by the sperm will take place near the end of this week.

**Week 3**  
Thirty hours after conception, the cell splits into two. Three days later, the cell (zygote) has divided into 16 cells. After two more days, the zygote has migrated from the fallopian tube to the uterus (womb). Seven days after conception, the zygote burrows itself into the plump uterine lining (endometrium). The zygote is now known as a blastocyst.

**Week 4**  
The developing baby is tinier than a grain of rice. The rapidly dividing cells are in the process of forming the various body systems, including the digestive system.

Figure 35 - Pregnancy Timeline

Figure 35 shows the *pregnancy timeline* which includes important information about the pregnancy process.

**Week 4**  
The developing baby is tinier than a grain of rice. The rapidly dividing cells are in the process of forming the various body systems, including the digestive system.

**Week 5**  
The evolving neural tube will eventually become the central nervous system (brain and spinal cord).

**Week 6**  
The baby is now known as an embryo. It is around 3 mm in length. By this stage, it is secreting special hormones that prevent the mother from having a menstrual period.

**Week 7**  
The heart is beating. The embryo has developed its placenta and amniotic sac. The placenta is burrowing into the uterine wall to access oxygen and nutrients from the mother's bloodstream.

**Week 8**  
The embryo is now around 1.3 cm in length. The rapidly growing spinal cord looks like a tail. The head is disproportionately large.

**Week 9**  
The eyes, mouth and tongue are forming. The tiny muscles allow the embryo to start moving about. Blood cells are being made by the embryo's liver.

**Week 10**  
The embryo is now known as a fetus and is about 2.5 cm in length. All of the bodily organs are formed. The hands and feet, which previously looked like nubs or paddles, are now evolving fingers and toes. The brain is active and has brain waves.

**Week 11**  
Teeth are budding inside the gums. The tiny heart is developing further.

**Week 12**  
The fingers and toes are recognisable, but still stuck together with webs of skin. The first trimester combined screening test (maternal blood test + ultrasound of baby) can be done around this time. This test checks for trisomy 18 (Edward syndrome) and trisomy 21 (Down syndrome).

**Week 13**  
The fetus can swim about quite vigorously. It is now more than 7 cm in length.

**Week 14**  
The eyelids are fused over the fully developed eyes. The baby can now merrily cry, since it has vocal cords. It may even start sucking its thumb. The fingers and toes are growing nails.

**Week 16**  
The fetus is around 14 cm in length. Eyelashes and eyebrows have appeared, and the tongue has tastebuds. The second trimester maternal serum screening will be offered at this time if the first trimester test was not done (see week 12).

**Week 18-20**

Figure 36 - Pregnancy Timeline

Figure 36 shows a brief description of the weekly growth of the baby.

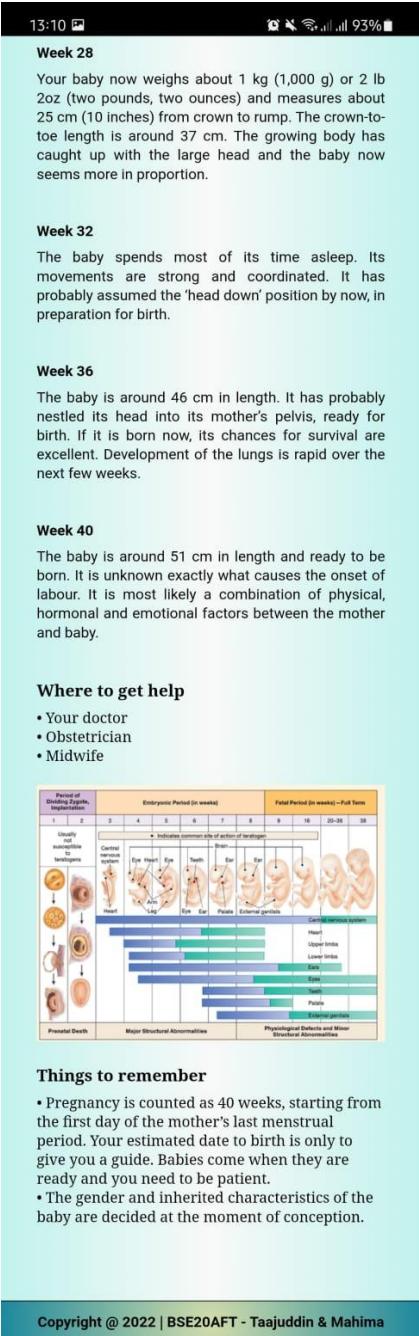


Figure 37 - Pregnancy Timeline

Figure 37 shows some important points to remember during pregnancy.

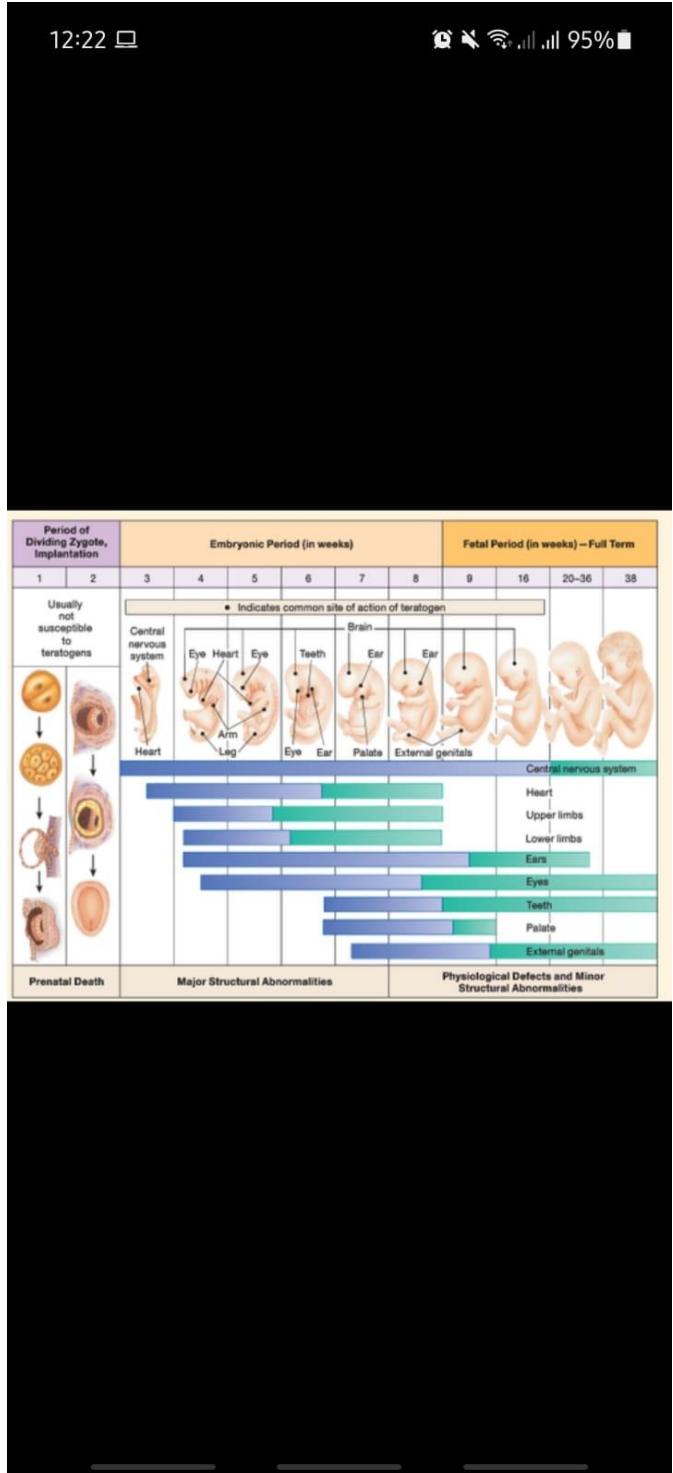


Figure 38 shows a chart of the development of the baby.

#### 7.4 My Bump (Homepage)

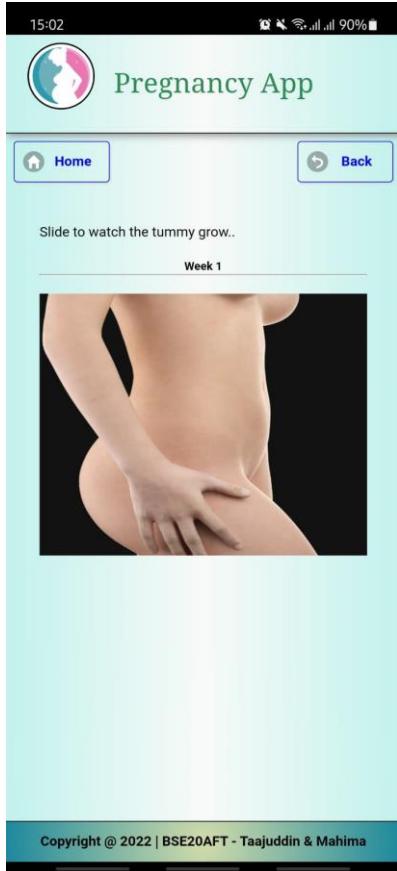


Figure 39 - My Bump

Figure 39 shows *My Bump* which consist of images of the belly evolution during the pregnancy process.

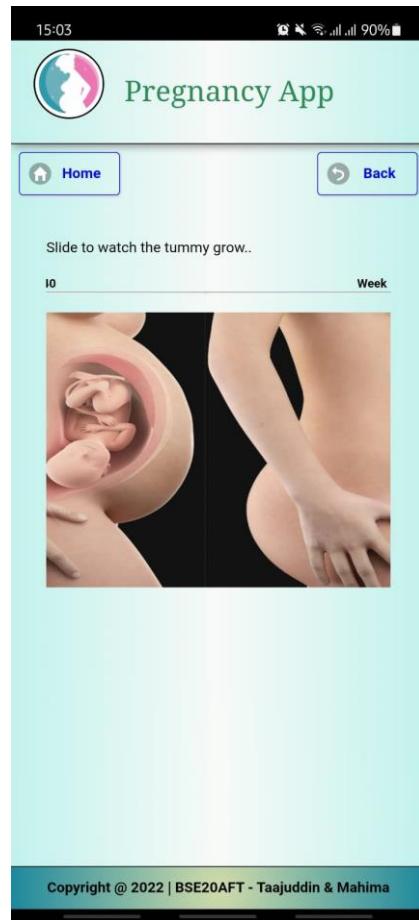


Figure 40 - My Bump

Figure 40 shows *My Bump* that allow a pregnant woman to see (by the swipe event) how the bump grows during these 40 weeks.

#### 7.5 Ultrasound Image (Homepage)



Figure 41 - Ultra Sound Image Page

Figure 41 shows *Ultra Sound Image* which comprise of the development of the baby.

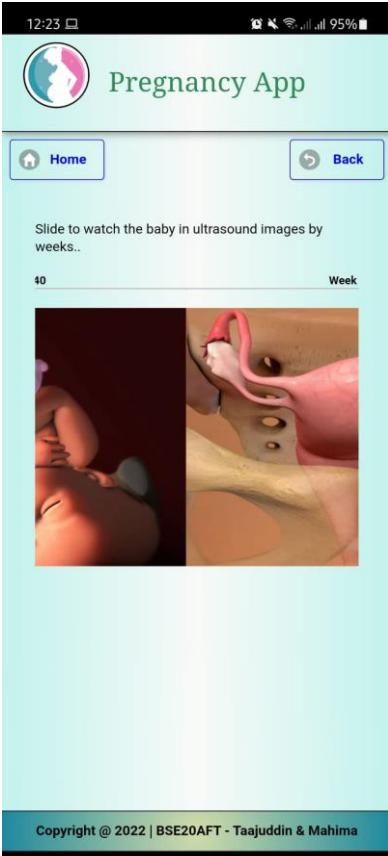


Figure 42 - Ultra Sound Image

Figure 42 shows *Ultra Sound Image* that allow the mother to see (by the swipe event) how the baby develop during these 40 weeks.

## 7.6 Symptoms Page



Figure 43 – Symptoms Page

Figure 43 shows *Symptoms Page* which consists of several symptoms a woman can experience during pregnancy.

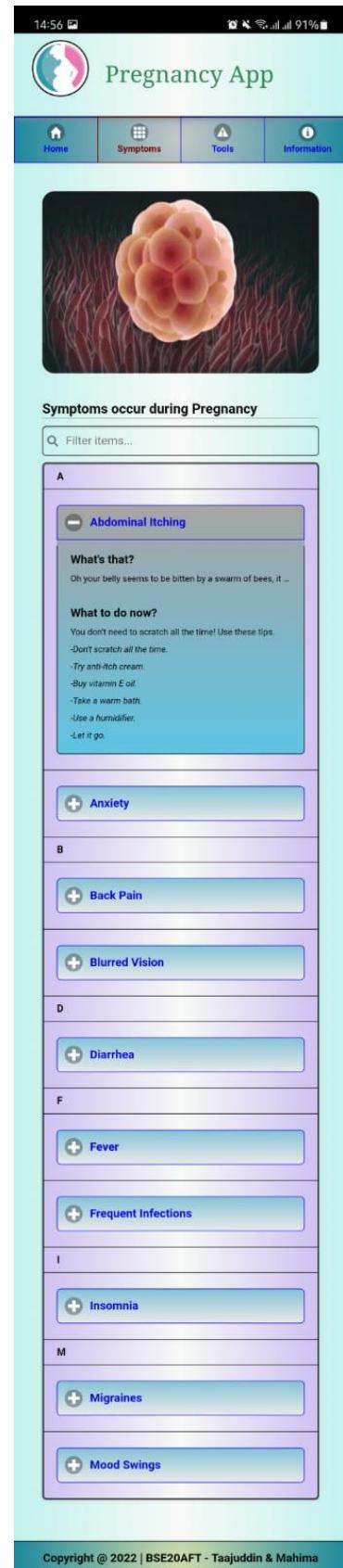
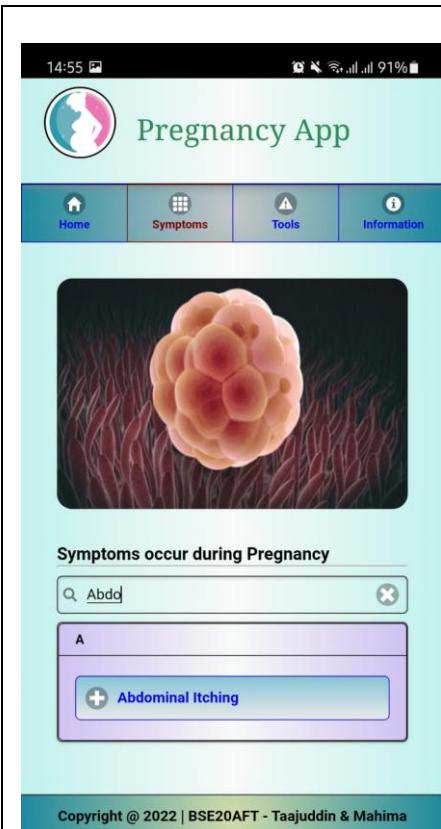


Figure 44 - Symptoms

Figure 44 shows *Symptoms Page* with the various symptoms along with some helpful tips.



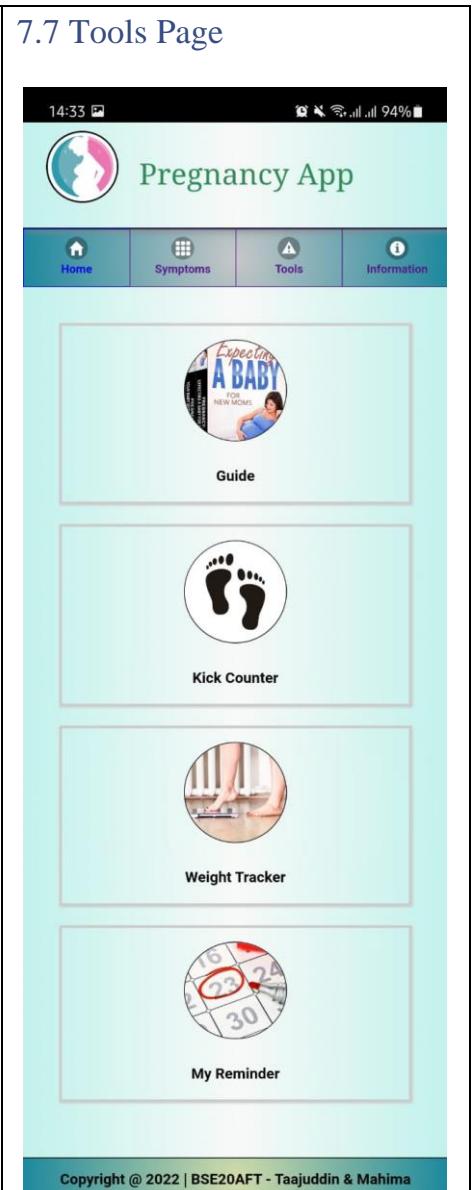
*Figure 45 – Symptoms*

Figure 45 shows a *search bar* that allow users to search for a specific symptom.



*Figure 46 - Symptoms*

Figure 46 shows that particular symptom along with its solution.



*Figure 47 – Tools Page*

Figure 47 shows the *Tools Page* where some functionalities has been implemented.

## 7.8 Guide (*Tools page*)



Figure 48 – Guide

Figure 48 shows *Guide* where users can obtain guidance related to their health and nutrition.



Figure 49 – Guide

Figure 49 shows various categories of nutrients that a pregnant woman should rely on during pregnancy.

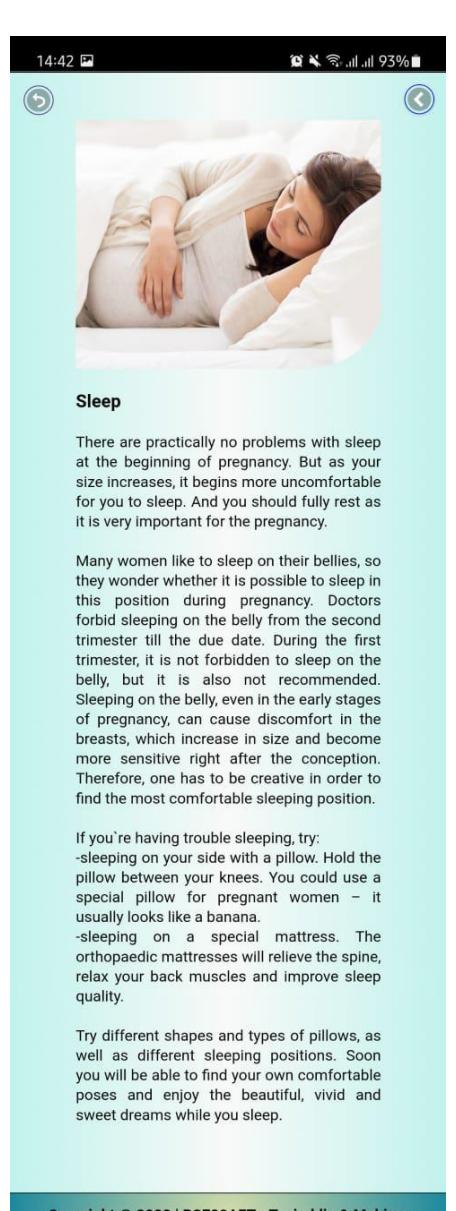


Figure 50 Guide

Figure 50 shows the benefits and practices of rest during pregnancy. User can click on any category to receive more information about other listview.

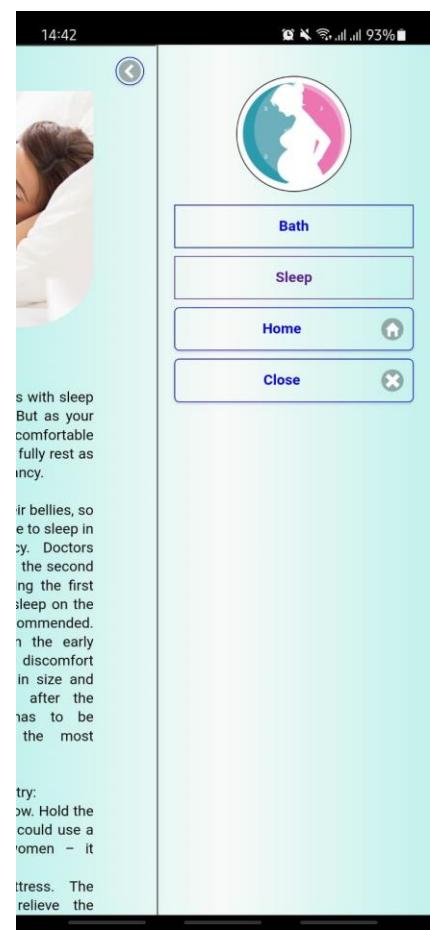


Figure 51 – Panel

Figure 51 shows a panel where user can get access to any category found in a particular listview.

## 7.9 Kick Counter (*Tools page*)

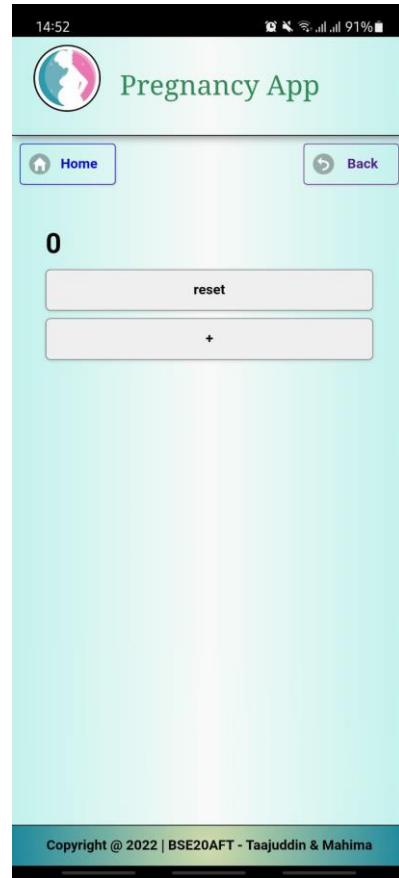


Figure 52 - Kick Counter

Figure 52 shows the kick counter where a pregnant woman can record the kick of the baby per day.

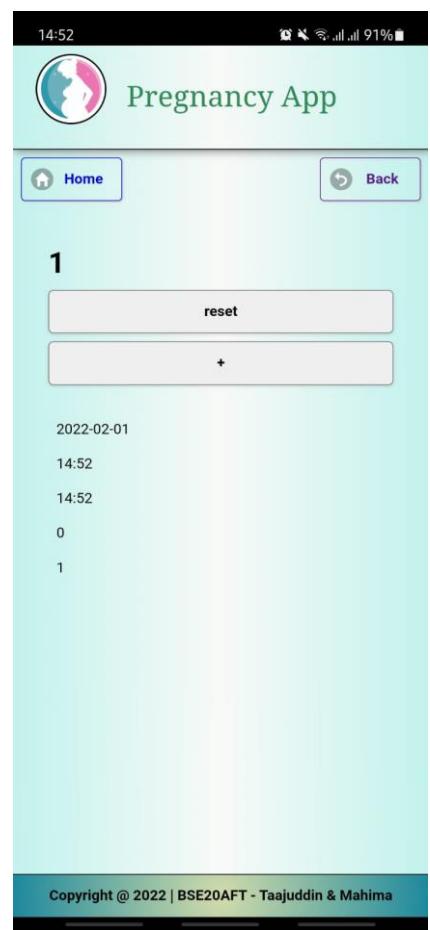


Figure 53 - Kick Counter

Figure 53 shows the result of one kick recorded. The actual date, start and finish time, and period is also recorded.

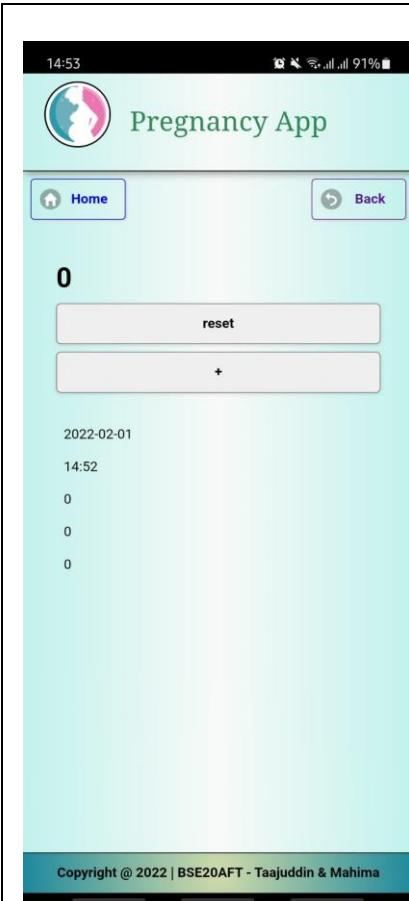


Figure 54 - Kick Counter

Figure 54 shows the reset button being used. The finish time, period and quantity are also reset.

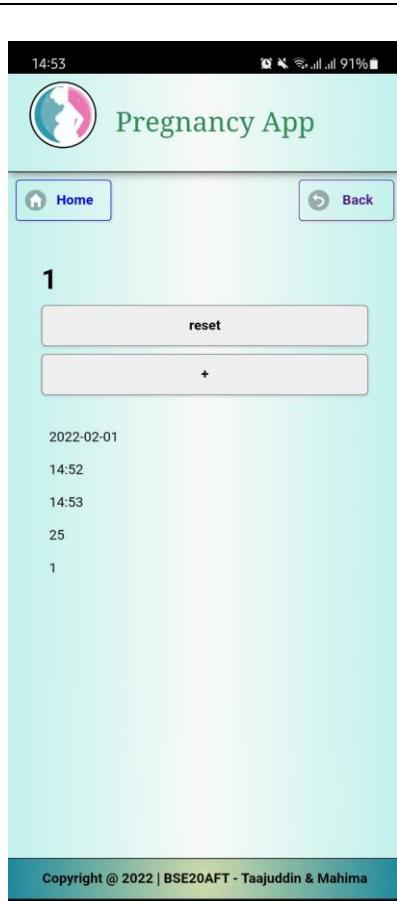
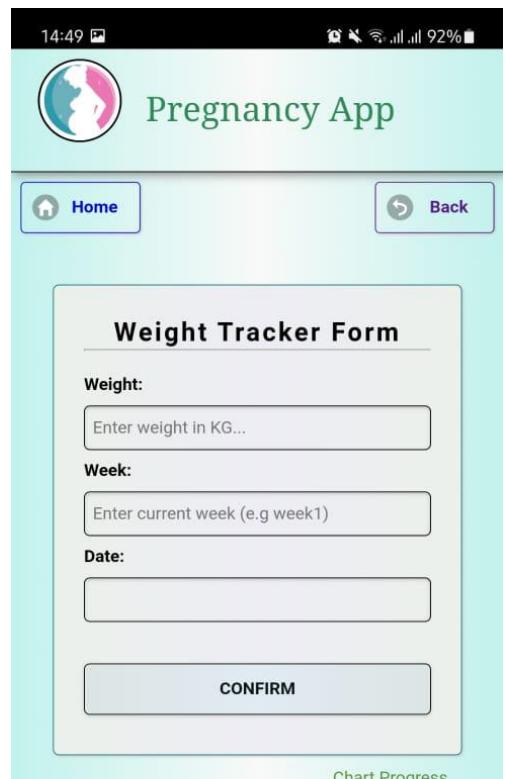


Figure 55 - Kick Counter

Figure 55 shows the user has re-started to count the kick of the baby.

## 7.10 Weight Tracker (Tools page)



[Chart Progress](#)

### Weight Tracker

Weight: 49 Week: Week1 Date: 2022-01-29
Weight: 49 Week: Week1 Date: 2022-01-30
Weight: 50 Week: Week1 Date: 2022-02-01
Weight: 51 Week: Week1 Date: 2022-02-04

Copyright @ 2022 | BSE20AFT - Taajuddin & Mahima

Figure 56 - Weight Tracker

Figure 56 shows the *weight tracker* where a pregnant woman can keep track of her weight. The user should enter the actual weight, week of pregnancy and date and it displays in a listview below.

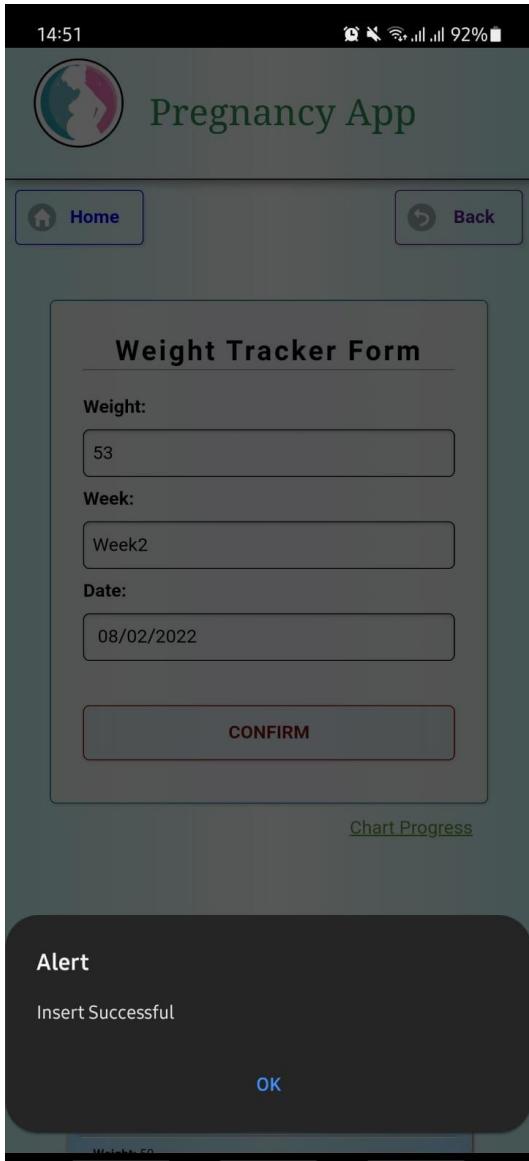


Figure 57 - Weight Tracker

Figure 57 shows once the information has been inserted in the database, an alert will be displayed on the screen.

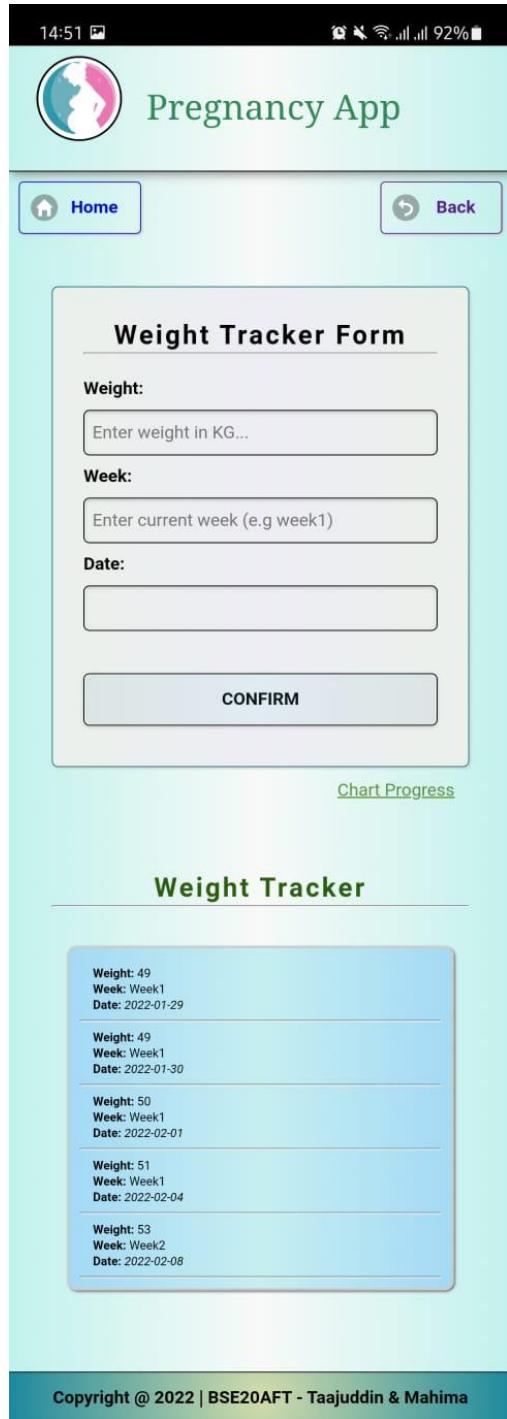


Figure 58 - Weight Tracker

Figure 58 shows the details that are recorded in the database are then displayed on a card below.



Figure 59 - Weight Chart

Figure 59 shows the Weight Chart of the recorded weight which is being displayed in the form of graph.

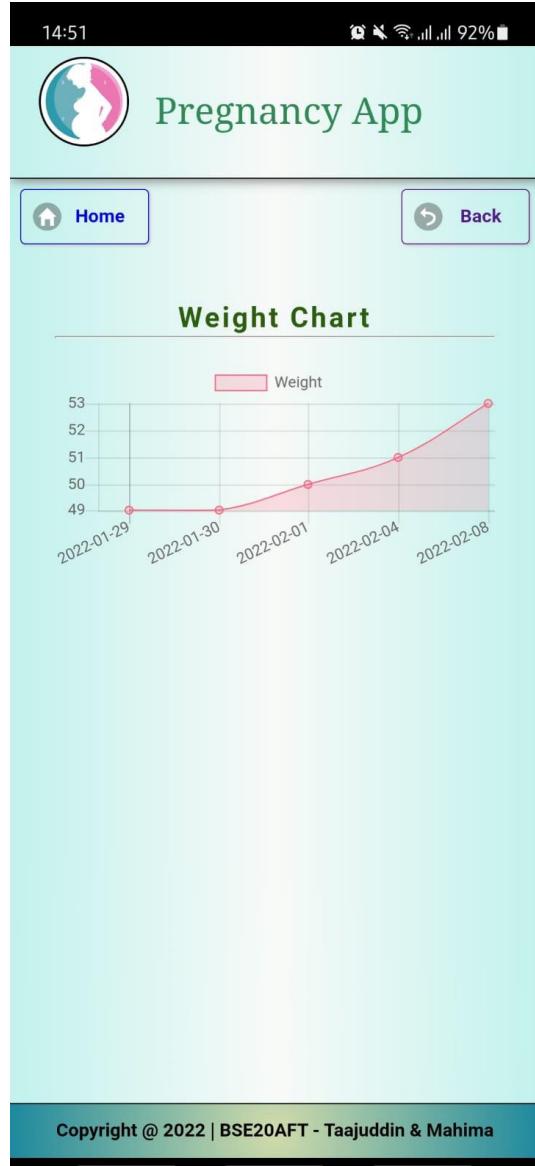


Figure 60 - Weight Chart

Figure 60 shows the Weight Chart of the recorded weight with a continuous growth as the woman's weight increases.

## 7.11 My Reminder (*Tools page*)

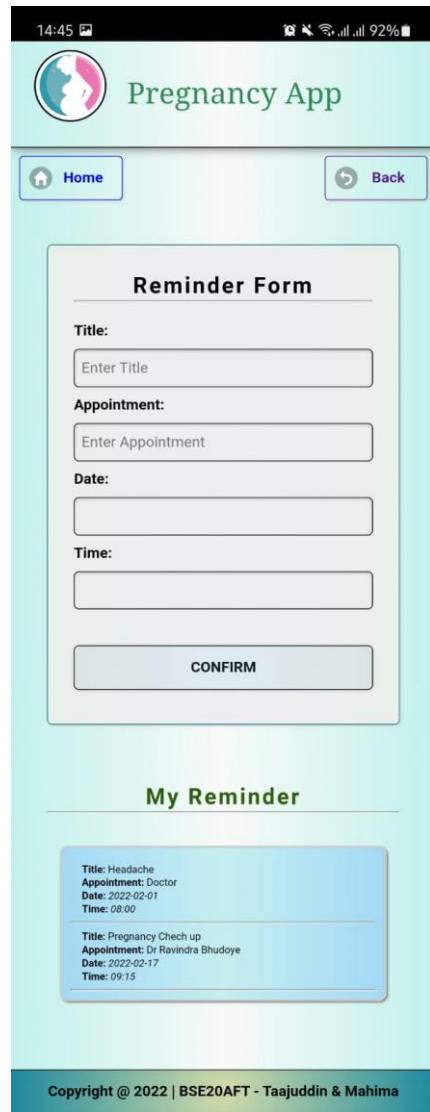


Figure 61 - Reminder

Figure 61 shows the *Reminder Form* where a pregnant woman can keep record of all her appointments. The user should enter the title, appointment, date and time.

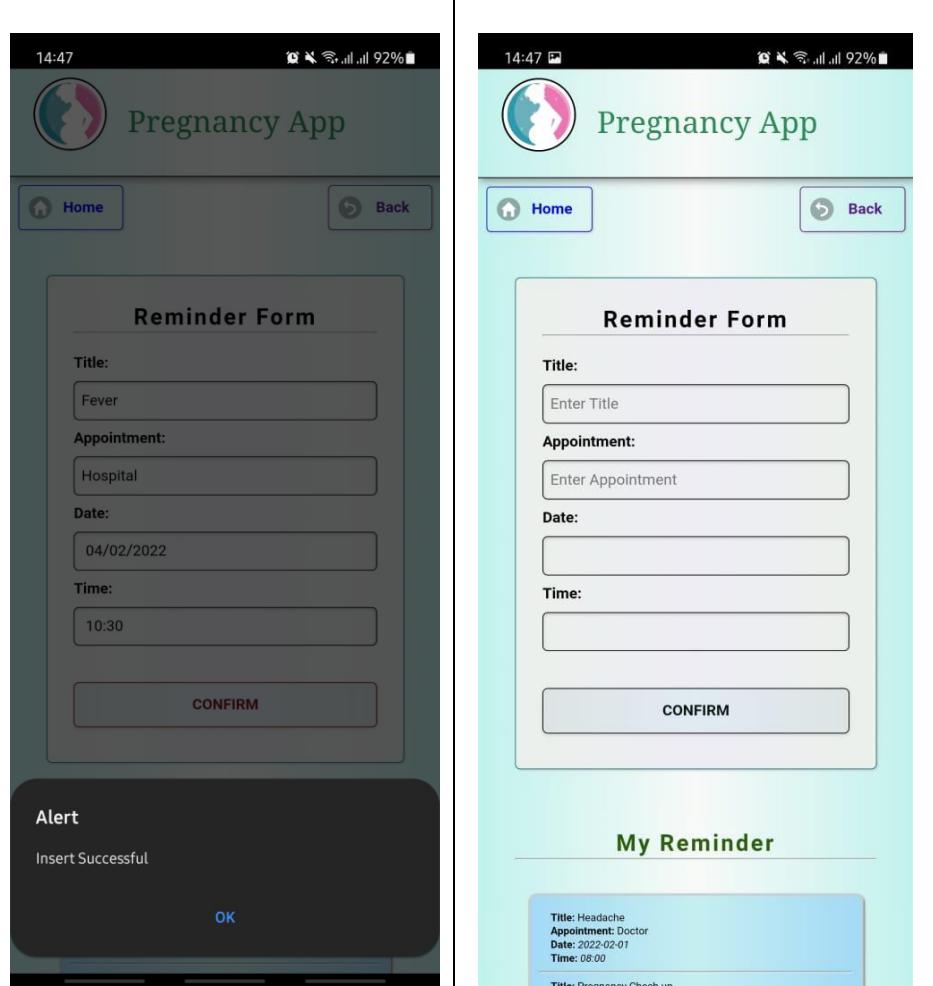


Figure 62 - Reminder

Figure 62 shows once the information has been inserted in the database, an alert will be displayed on the screen.

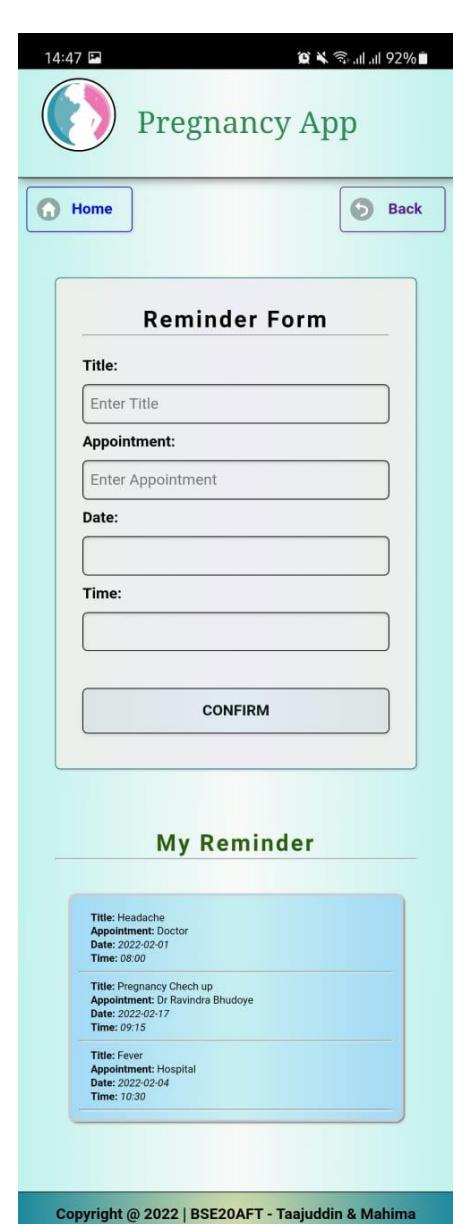


Figure 63 - Reminder

Figure 63 shows the details that are recorded in the database are then displayed on a card below.

## 7.12 Information Page



## 7.13 About App (*Information page*)

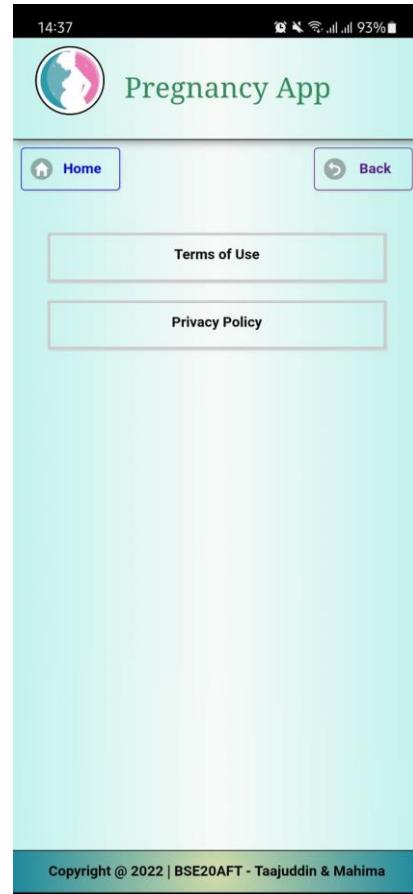


Figure 65 shows the About App with consist of Terms of use and Privacy Policy.

Figure 64 shows the Information Page with some functionality.

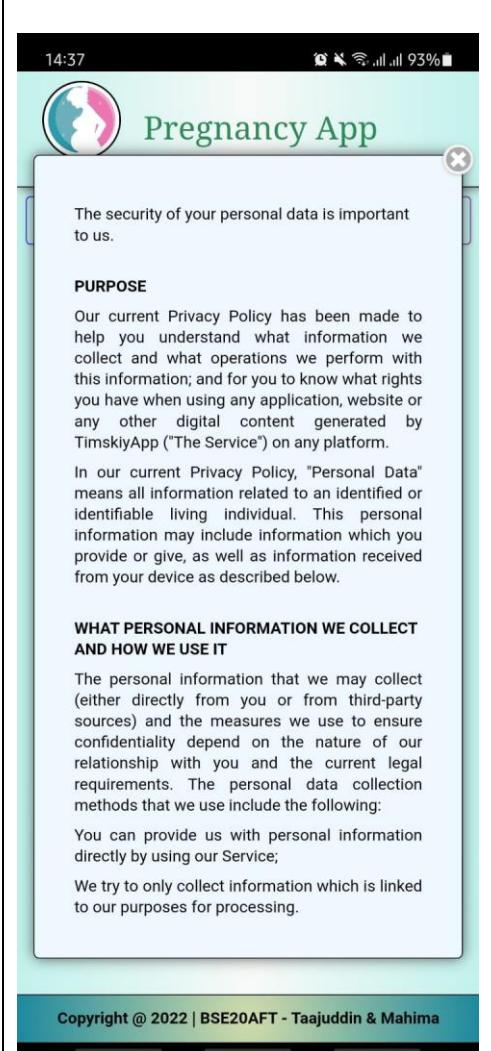


Figure 66 shows the Privacy Policy where the security of personal data is explained to the user.

## 7.14 Contact Us (Information page)

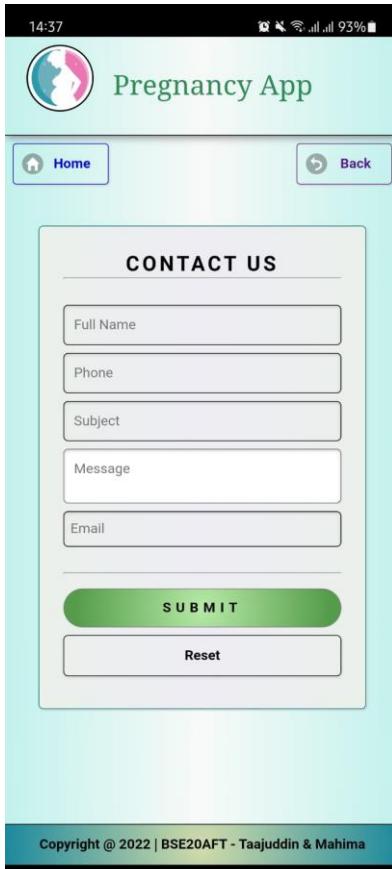


Figure 67 - Contact Us

Figure 67 shows the *Contact Us Form* where a pregnant woman can send message in case there is any query. The user should enter the full name, phone, subject, message and Email.

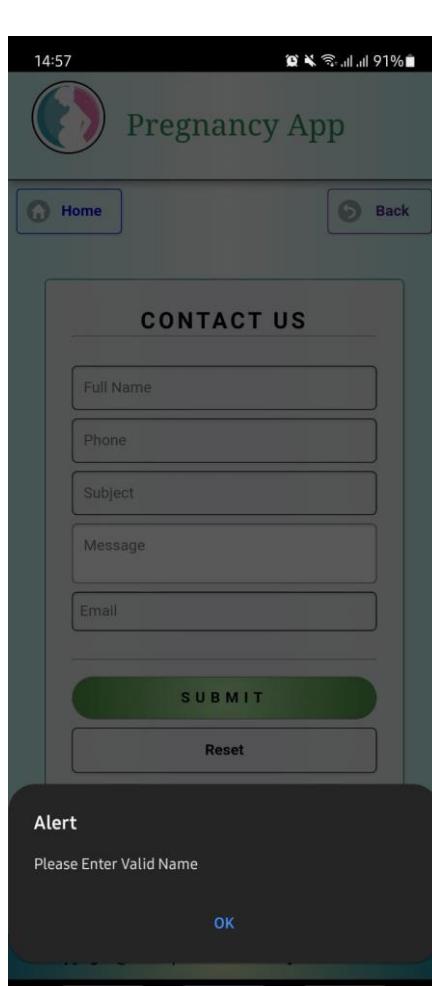


Figure 68 - Validation

Figure 68 shows validation for name. In case a valid name has not been entered, an alert will be displayed on the screen.

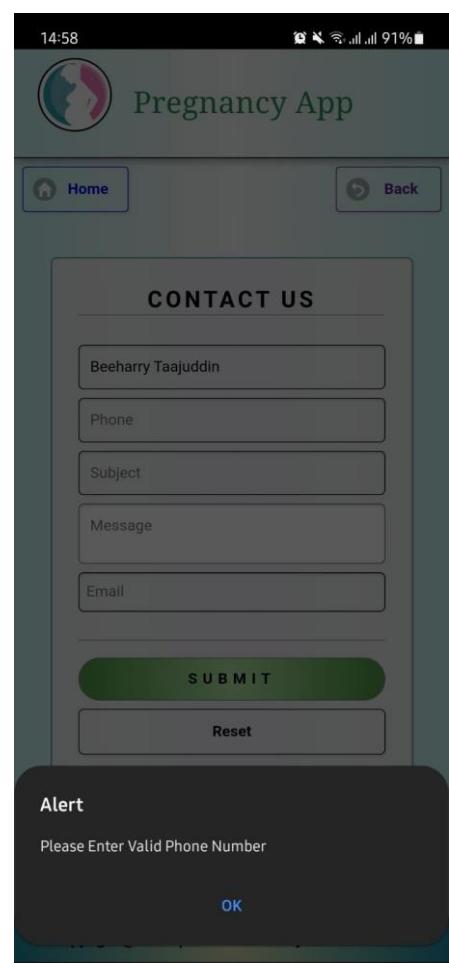


Figure 69 - Validation

Figure 69 shows validation for phone number. In case a valid phone number has not been entered, an alert will be displayed on the screen.

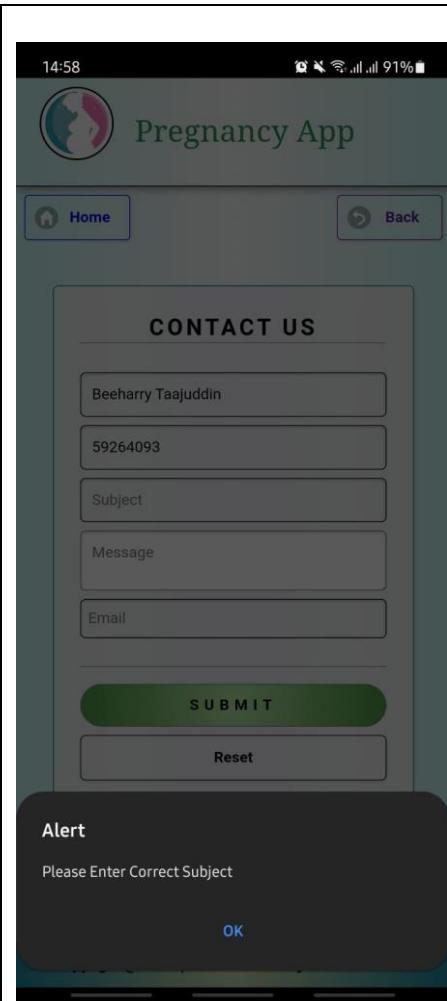


Figure 70 – Validation

Figure 70 shows validation for subject. In case a valid subject has not been entered, an alert will be displayed on the screen.

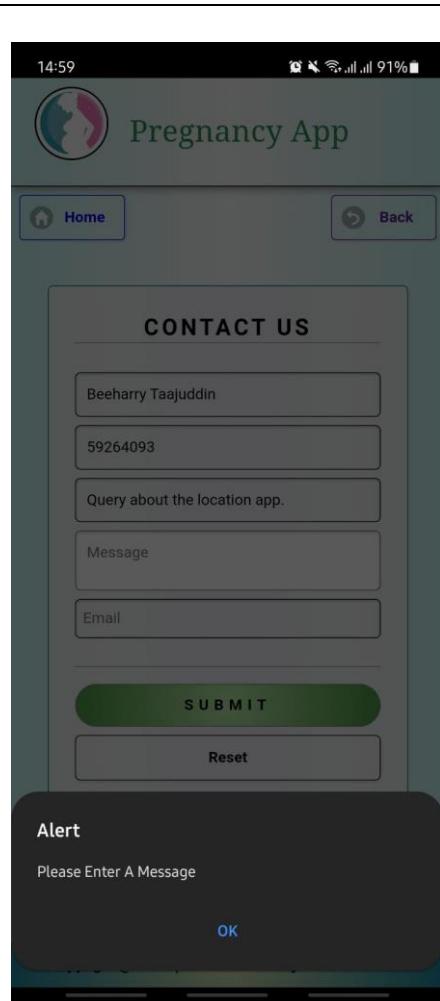


Figure 71 - Validation

Figure 71 shows validation for message. In case the message textbox has not been filled, an alert will be displayed on the screen.

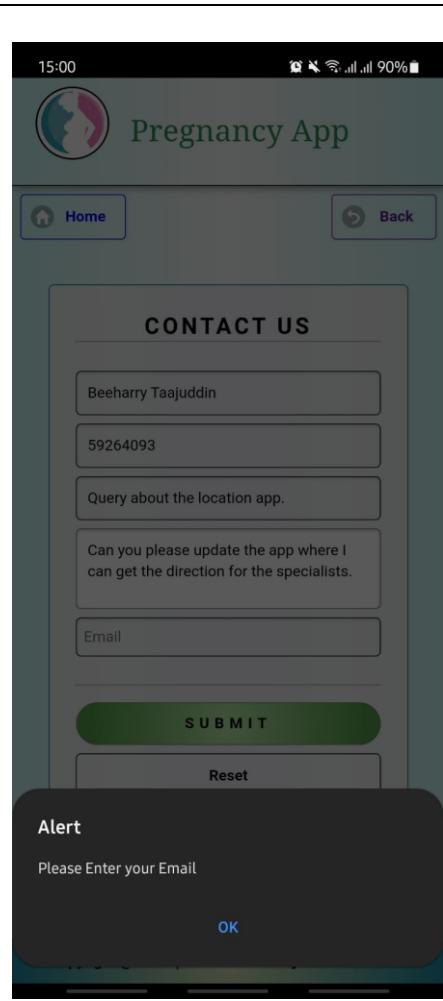


Figure 72 - Validation

Figure 72 shows validation for email. In case the email textbox has not been filled, an alert will be displayed on the screen.

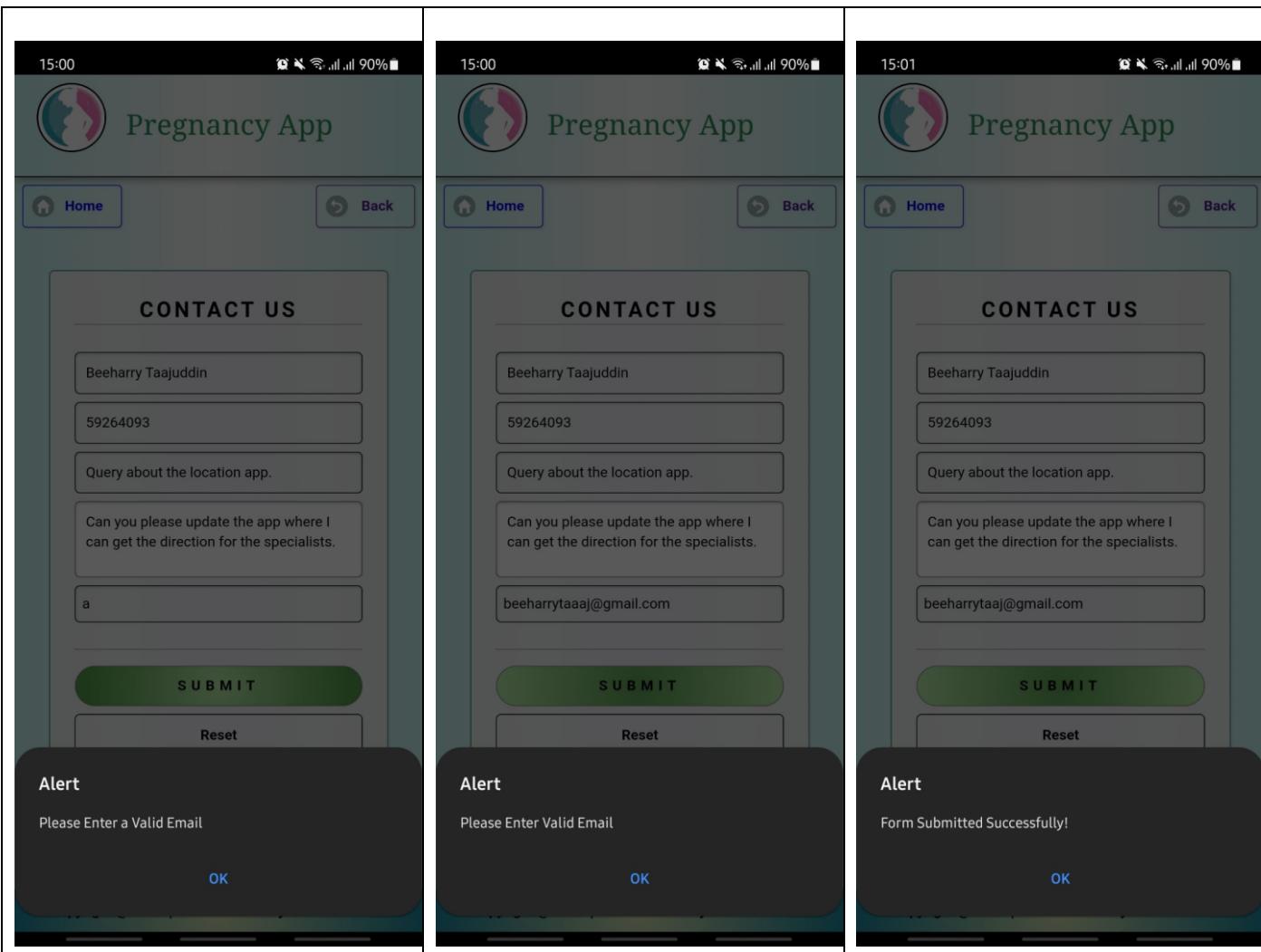


Figure 73 – Validation

Figure 73 shows validation for email. In case a valid email has not been entered, an alert will be displayed on the screen.

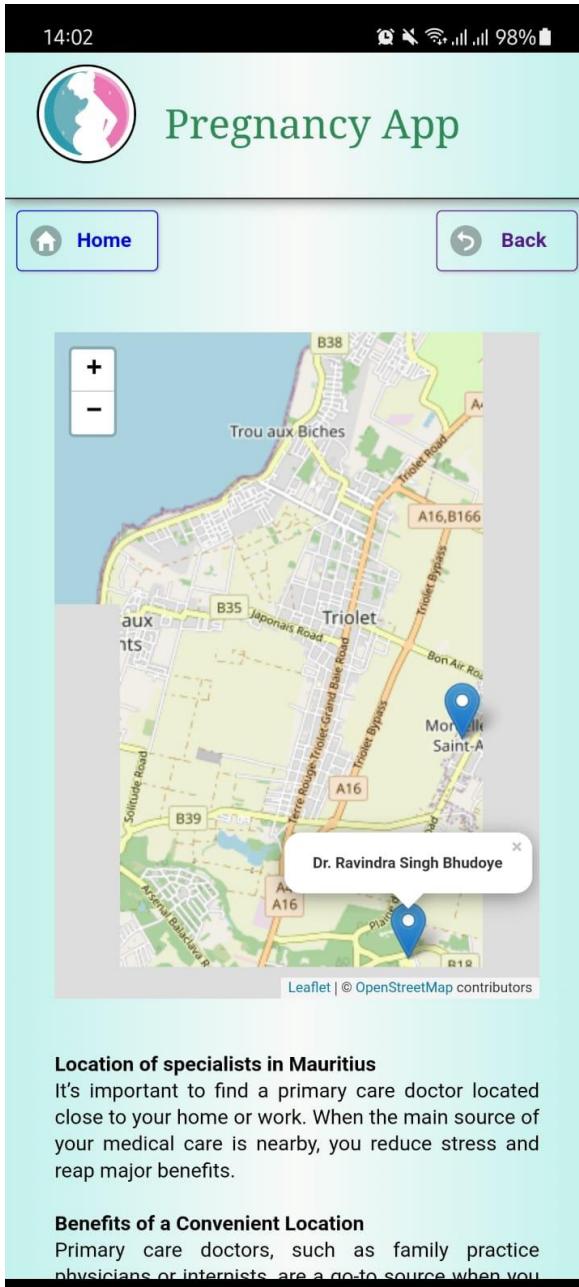
Figure 74 - Validation

Figure 74 shows validation for email. Once the correct email has been entered, click on *ok*.

Figure 75 - Validation

Figure 75 shows when the form has successfully been submitted, an alert will be displayed on the screen.

## 7.15 Location of Specialists (*Information page*)



### Location of specialists in Mauritius

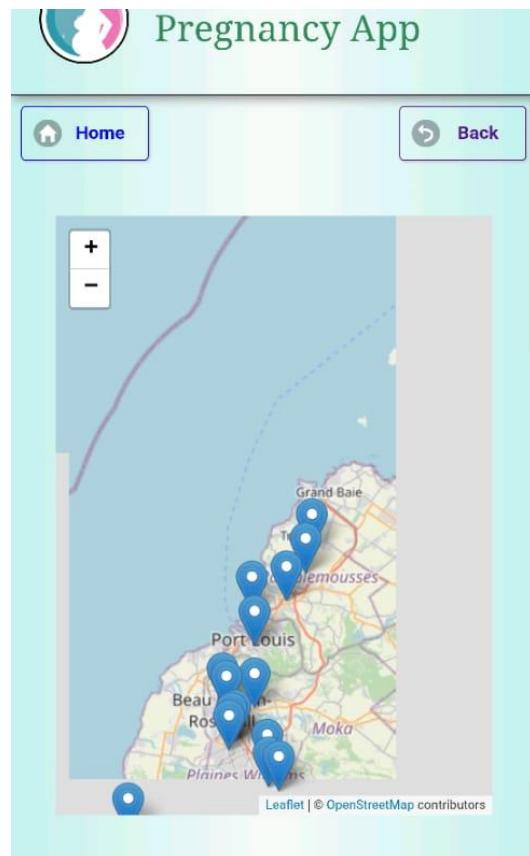
It's important to find a primary care doctor located close to your home or work. When the main source of your medical care is nearby, you reduce stress and reap major benefits.

### Benefits of a Convenient Location

Primary care doctors, such as family practice physicians or internists, are a go-to source when you

*Figure 76 – Location (1)*

Figure 76 shows the location of different specialists found in Mauritius.



### Location of specialists in Mauritius

It's important to find a primary care doctor located close to your home or work. When the main source of your medical care is nearby, you reduce stress and reap major benefits.

### Benefits of a Convenient Location

Primary care doctors, such as family practice physicians or internists, are a go-to source when you need medical care. They handle routine illnesses, manage chronic diseases, and provide vital wellness services; so it makes life a lot easier if they are in a location that's convenient for you. Here are three reasons location matters:

#### Less Time in Traffic

Finding a doctor close to your home makes it easy to fit in a visit on the way to or from work, and reduces the drive from home when you feel really sick. Likewise, having a doctor close to work can lessen time away from the office when you need a visit.

#### Address Health Concerns Promptly

If seeing the doctor involves a drive across town or a commute, you're more likely to put off health issues. It's best to address medical concerns promptly when they can be more easily treated.

#### Follow-Up is Easy

If your doctor recommends follow-up care, it's much more convenient if the office is close by. You'll be more likely to stick with appointments for any ongoing care you may need.

*Figure 77 – Location (2)*

Figure 77 shows the benefits of location. User can get the location of specialists in the map shown above.

## 7.16 Current Location (*Information page*)

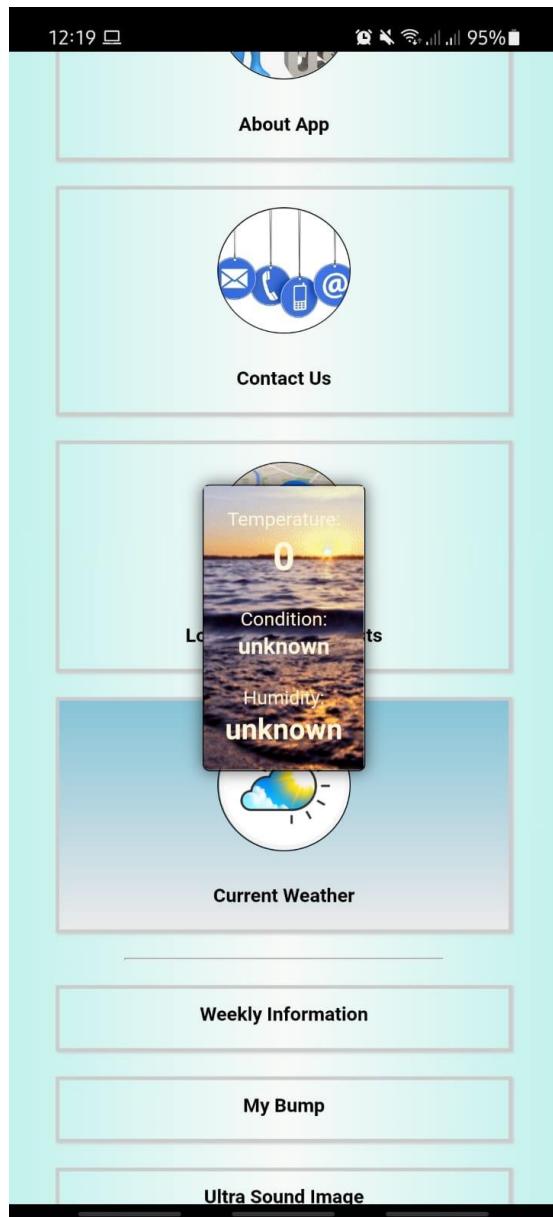


Figure 78 - Current Location

In figure 78, it should depict the current location of the user but unfortunately the weather forecast does not display on the android application even though all permissions were implemented.