**ASSETS REQUIRED:**

1. PRODUCT NAME (MADEUP NAME – THIS SOLUTION IS NOT TO BE USED IN PRODUCTION).
2. IMAGE ASSETS, IF REQUIRED
3. TEXT CONTENT IF REQUIRED
4. LOGO (IF UNAVAILABLE, A REQUEST FOR LOGO-DESIGN COULD BE PLACED), IF REQUIRED
5. FYP ABSTRACT
6. FYP REPORT (IF AVAILABLE)

**HARDWARE REQUIRED:**

1. RASPBERRY PI ZERO.
2. RASPBERRY PI CAMERA.
3. SD CARD.
4. ZONG 4G DEVICE.
5. ASUS 3000 MAH BATTERY.
6. HALL EFFECT SENSOR MODULE.
7. WIRELESS CHARGING MODULE.
8. NODE MCU.
9. NEO 6M-V2 GPS MODULE.
10. BUCK/BOOST CONVERTERS FOR CHARGING DISCHARGING
11. BATTERY CHARGING MODULES.
12. WALLET / PURSE.
13. MISC ITEMS (PCB / BREADBOARD/JUMPERS ETC)

**\*** Some of theassets mentioned above would not be utilized, and therefore returned. The client is required to drop off and collect all of the assets at the locations prescribed.

**\*\*** In case of components burning out or malfunctioning, the client would be required to provide new components. The damaged hardware would be returned.

**\*\*\*** If any of the hardware components get damaged by the developers, technicians or engineers in a case where the atrocity could have been **easily** averted, the components would then be replaced by our team. The damaged hardware would be returned.

**MILESTONES ACHIEVED (CLIENT):**

1. PROJECT BATTERY MANAGEMENT.
2. WIRED + WIRELESS CHARGING.
3. WIFI (WLAN) HOTSPOT.
4. ~~GPS TRACKING. (WITH BLYNK APP ONLY).~~ (WILL BE CONSIDERED)

**MILESTONES ACQUIRED (CLIENT):**

THE ANDRIOD APPLICATION THAT IS REQUIRED WILL HAVE TWO GUI BUTTONS,

1. ~~GPS LOCATION POWERED THROUGH GOOGLE MAP JUST LIKE BLYNK APP FROM THE TUTORIAL. USING THE CODE PROVIDED IN THE TUTORIAL. ).~~ (WILL BE CONSIDERED)
2. CAMERA IMAGES IN LOST MODE.

**LOST MODE CRITERIA. (ANTI THEFT) – FEATURE SPECIFICATION**

A ZONG 4G DEVICE IS PLACED INSIDE THE PURSE POWERED BY PROJECT BATTERIES TO THE PCB WITH A BUTTON.

1. ~~FOR LOST MODE, THE MICROCONTROLLER WILL CONNECT TO UR CELL PHONE USING BLUETOOTH.~~

FOR THE ACTIVATION OF LOST MODE, EITHER:

**YOUR PHONE WILL CONNECT TO YOUR WALLET THROUGH WIFI (WLAN).**

**OR**

YOUR WALLET/CONTROLLER WILL CONNECT TO YOUR PHONE THROUGH WIFI (WLAN). THE LATTER IS TRICKY AND LESS REDUNDANT SINCE SMARTPHONES PROVIDE A BETTER INTERFACE TO CONNECT TO NEARBY DEVICES.

1. IF THE CONNECTION ON THE ~~BLUETOOTH~~ WIFI (WLAN) BREAKS, A PUSH NOTIFICATION SHOULD BE SENT TO PHONE TO ACTIVATE THE LOST MODE.
2. IF FOR (CERTAIN) TIME THE USER DID NOT RESPOND TO THE LOST MODE,

THE LOST MODE (PHASE II) IS ACTIVATED (ON WALLET).

1. THE LOST MODE WILL FIRST ACTIVATE THE HALL EFFECT SENSOR PLACED ON THE OPENING LIP OF THE WALLET OR PURSE THAT WILL BE USED AS A TRIGGER FOR THE NIGHT VISION CAMERA, PLACED INSIDE THE WALLET.
2. THE HALL EFFECT SENSOR, UPON A BREAK IN CONNECTION, WOULD USE THE CAMERA TO CAPTURE PICTURES AT THE RATE OF 1 IMAGE PER SECOND (MAX OUTPUT FOR A STREAM, IF TO BE GENERATED, CANNOT BE THUS GREATER THAN 1FPS) FOR A MINUTE OR TWO.
3. THOSE PICTURES, THEN WOULD BE UPLOADED TO THE CLOUD THROUGH THE 4G DEVICE DATA PACKAGE PLACED INSIDE WALLET.
4. THE ANDRIOD APP CONNECTS TO REMOTE (CLOUD) SERVER AND DOWNLOADS IMAGES TO BE THEN VIEWED ON PHONE.

ANY CRITICISM IS WELCOME ALONG WITH QUERIES AS WELL.

**DELIVERABLES (SLABS) - MILESTONES:**

1. **RasPI (OS + Environment) Initiation**:
   1. INSTALL RASPBIAN OS ON THE **RASPBERRY PI ZERO**.
   2. ENABLE *VNC, CAMERA, SSH, SPI, I2C* INTERFACES ON THE RASPI.
   3. SETUP THE PI TO CONNECT TO MOBILE HOTSPOT DEVICE
   4. IF POSSIBLE, SETUP THE *AP*. TO OVERWRITE THE *DHCP’s* TABLE AND ASSIGN STATIC IPS TO CONNECTED DEVICES.
   5. INSTALL AND CONFIGURE **NODEJS** AND **NPM** ON THE RASPI.
   6. SETUP RASPI STARTUP SCRIPTS TO INITALIATE **NODEJS** SERVER.’
   7. DOWNLOAD AND INSTALL REQUIRED PACKAGES AND DEPENDENCIES ON RASPI.
2. **Wallet Client (Node Development)**
   1. CONFIGURE THE **NODEJS**-INSTANCE TO BYPASS THE KERNEL, AND ACCESS ROOT-LEVEL PERMISSIONS
   2. SETUP A ROUTING MIDDLEWARE CLIENT ON THE PI TO CATER TO HTTP REQUESTS OVER LAN
   3. SETUP A WEB-SOCKET TO CONNECT TO CLOUD-HOST FOR MESSAGES & NOTIFICATIONS FORWARDING.
   4. CONFIGURE THE INSTANCE TO ACCESS I/O PINS, DATA & COMM LINES AND OPEN PORTS.
   5. INITIALIZE THE SERVER TO LISTEN ON A SPECIFIC PORT.
   6. OPEN A PIPE TO ENABLE I2C BASED COMM WITH GPS
   7. OPEN A PIPE TO ENABLE I2C BASED COMM WITH PICAMERA
   8. SETUP I/O PROCESSES TO INTERACT WITH SENSORS/MODULES.
   9. COMPLETE THE MUTUAL REQUIREMENTS FOR THIS SLAB:
      1. Continually try to connect to the AP on initialization, until connected
      2. Accept all socket-based connections from Android Applications
      3. Using Google Maps API, WiFi loclization GPS, update the wallet’s location with minimal interval on the cloud
      4. Upon closing of sockets, if the connected devices zero-out, send an HTTP POST request to the API, live on the cloud for a ‘Wallet lost’ alert.
      5. For a certain amount of time, check the cloud-instance if the user approved of the notification, if not, immediately activate the LOST-MODE.

This activates the hall-effect to capture images when the circuit is tripped. The camera is to filter/capture 1 frame per second.

The images will then be forwarded to the cloud to be saved in cloud-storage.

1. **Circuit Design (Sensors Integration)**
   1. COLLECT DATASHEETS FOR THE SENSORS TO BE INTEGRATED
   2. DESIGN AND CONSTRUCT DRIVER / CONTROLLER / LOGIC-CONVERTER CIRCUITS FOR EACH MODULE, IF REQUIRED
   3. HOOK UP THE MODULES TO RESPECTIVE (PREVIOUSLY ASSIGNED I/O PINS), FOR STREAMING *NRT* DATA.
2. **Breakout-board Modules Integration**
   1. COLLECT DATASHEETS FOR THE MODULES TO BE INTEGRATED
   2. DESIGN AND CONSTRUCT DRIVER / CONTROLLER / LOGIC-CONVERTER CIRCUITS FOR EACH MODULE, IF REQUIRED
   3. HOOK UP THE MODULES TO RESPECTIVE (PREVIOUSLY ASSIGNED I/O PINS), FOR STREAMING *NRT* DATA.
   4. SET UP THE MASTER TO CONTROL EACH OF THE MODULES, AND BEGIN RX/TX COMM
   5. PARSE AND SETUP DATA-PROCESSORS FOR BOTH THE SENSORS, AND BREAKOUT-BOARD MODULES
3. **CAMERA STREAM I/O + BROADCAST**
   1. I2C WTH THE RASPI TO TRANSMIT AS A SLAVE
   2. STAY IN DISABLED MODE ON DEFAULT
   3. FORWARD THE BEST QUALITY ENCODED MEDIA BUFFERS TO THE WALLET-INSTANCE. THE CAPTURE-EVENT WILL FIRE FOR A DURATION OF TWO MINUTES.
4. **ANDROID APPLICATION DEVELOPMENT**
   1. INITIALIZE AN ANDROID PROJECT WITH TARGET API LEVEL OF 28 AND BASE OF 19
   2. INTEGRATE MATERIAL DESIGN – PREFERABLE DARK THEME
   3. SET UP PROJECT ASSETS AND VARIABLES
   4. SHOW THE STATUS OF NODE ON THE NETWORK (WALLET – CONNECTED/ DISCONNECTED/ CONNECTING)
   5. ON THE HOME PAGE, A MAP SHOULD BE STRETCHED SO THAT THE CURRENT LOCATION OF THE WALLET IS AVAILABLE (NOTE, NON-INDUSTRIAL GRADE MODELS HAVE AN ERROR RANGE OF UPTO 20metres.
   6. THERE WOULD BE A SECTION AVAILABLE TO VIEW IMAGES AVAILABLE ON THE CLOUD, LABELLED BY DATE AND TIME.
   7. ANDROID BACKGROUD SERVICE WILL CONNECT TO THE SOCKET ENABLED CLOUD HOST AND RECEIVE NOTIFICATIONS ON THIS CHANNEL
5. **API DEVELOPMENT (Cloud)**
   1. THE CLOUD API WOULD EITHER BE DEPLOYED ON **AWS, GCP** OR **HEROKU**. IF OPTING FOR AMAZON WEB SERVICES OR GOOGLE CLOUD PLATFORM, A HOLDER WITH CARD (PREFERABLY **MASTERCARD)** AND A ‘AVAILABLE BALANCE’ OF **$1** WOULD BE REQUIRED TO COMPLETE THE SUBSCRIPTION. A SUBSCRIPTION, FREE FOR THE FIRST YEAR WOULD LODGED.

THIS SERVICE CANNOT BE AVAILED BY OUR TEAM, SINCE ALL OF PAYMENT METHODS HAVE ALREADY EXHAUSTED THE **FREE-TIER** IN DEVELOPMENT**.**

* 1. SETUP THE WEB-SOCKET HOST ON THE CLOUD TO ENABLE THE WALLET-NODE AS WELL AS THE MOBILE APPLICATION TO CONNECT AND POST/ RECEIVE MESSAGES.
  2. SETUP A ROUTING MIDDLEWARE TO CATER TO FILE TRANSFERS AS WELL AS HTTP REQUESTS FOR EACH ROUTE.
  3. INCOMING FILES (IMAGES) WOULD BE SAVED TO CLOUD STORAGE
  4. THE SYSTEM WOULD NOT HAVE A DATABASE AND THUS A VOLATILE EVENTS MAP, BUT WILL BE *NRT* (NEAR REAL-TIME)

1. **API DEVELOPMENT (Cloud)**
   1. THE ENTIRE NETWORK AND SYSTEM WOULD INITALLY BE DEPLOYED LOCALLY, AND STAY AS SUCH TILL THE APPLICATIONS WOULD STAY IN DEVELOPMENT PHASE.
   2. THE LAST SLAB WOULD BE DEPLOYED AFTER CONFIRMATION OF THE CLIENT
   3. CLIENTS WOULD BE PROVIDED ADMIN LEVEL ACCESS TO UTILITIES, TOOLS AND TOOLS
   4. THE CLIENT WOULD BE PROVIDED WITH THE SOURCECODE, BUILDS, AND NECESSARY MATERIAL, HOWEVER **ALL OF THE SOURCECODE AND ASSETS AND THEIR WILL AUTOMATICALLY BE DELETED AFTER A PERIOD OF 3 MONTHS FROM EVERY LOCATION COPED TO.**

**DELIVERABLES (SLABS) & COSTING:**

* Each slab extend from 3 days to over a week. The deliverables belong to a slab each.
* Exchange of resources/Assets will only take place between slabs.
* The costs mentioned, are subject to change (due to requests or additions only).
* Major **change & addition requests** will be charged.
* **Requests** could alter duration of slabs and deliverables.
* Major **Change requests** will not be catered after the completion of deliverable they alter. The respective deliverable will be re-built if it has been completed.
* **Addition requests** could be extremely costly, and time-consuming.
* The project payment would be deployed in the following slabs:
  + 50% before initiation
  + 10% after completion of each slab
  + If slabs are delayed by a week, development would be paused, the deployment date would be considered the new date
  + If a request is to be catered in the current slab, or immediately catered due to whatever may be the cause, the request charges would be bound to immediately be released to adhere the request within
* This report is only valid for 1 day and would be revised if catered to later than than that**.**
* This project is to be used in an FYP only. All server-side executables & binaries, application builds, data & copies would be deleted automatically exactly **3 months** after project deployment. It is not to be used in production version, or to be redistributed.
* If either of the computation instances is migrated, or installed upon another system, our team will immediately be notified upon network availibility (this step to ensure that the solution isn’t being violated)

|  |  |  |  |
| --- | --- | --- | --- |
| DELIVERABLE | DURATION - day(s) | COST – pkr | SLAB |
| RasPI (OS + Environment) Initiation | - | 2,000 | 1 |
| Wallet Client (Node Development) | - | 6,000 | 2 |
| Circuit Design (Sensors Integration) | - | 1,500 | 1 |
| Breakout-board Modules Integration | - | 0 | 1 |
| CAMERA STREAM I/O + BROADCAST | - | 4,500 | 2 |
| ANDROID APPLICATION DEVELOPMENT | - | 14,000 | 3 |
| API DEVELOPMENT (Cloud) | - | 3,500 | 4 |
| 3rd PARTY APIs – INTEGRATION (per API / package) | - | 2,500 | 5 |
| DEPLOYMENT & DevOps | - | 2,000 | 5 |
|  |  |  |  |
|  |  |  |  |
| TOTAL |  | 36,000 |  |
|  |  |  |  |