Diario di lavoro

|  |  |
| --- | --- |
| Luogo | SAM Trevano |
| Data | 18.03.2019 |

|  |
| --- |
| Lavori svolti |
| Inizialmente mi sono occupato di portarmi avanti con la configurazione del Rspberry PI, una volta ricollegato e avviato ho configurato l’interfaccia statica *eth0* con l’indirizzo IP che mi era stato liberato due settimane fa.  Interfaccia eth0  auto lo  iface lo inet loopback  auto eth0  iface eth0 inet static  address 10.20.4.90  netmask 255.255.254.0  gateway 10.20.5.254  dns-nameservers 10.20.4.2  Una volta fatto ciò ho commentato le righe presenti nel file *apt.conf* corrispondenti al proxy del server, in modo tale da potermi collegare con successo ad internet. In seguito, tramite una guida (<https://blog.vinczejanos.info/2016/08/31/install-openalpr-on-raspberry-pi-3/>), ho cominciato ad installare tutti i requisiti che servono alla libreria di riconoscimento delle targhe. Il primo passo è quello di scaricare i pacchetti necessari, per farlo ho utilizzato un comando solo che contiene tutti i pacchetti:  apt-get install autoconf automake libtool libleptonica-dev libicu-dev libpango1.0-dev libcairo2-dev cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev python-dev python-numpy libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev virtualenvwrapper liblog4cplus-dev libcurl4-openssl-dev  In seguito ho scaricato le dipendenze da github nel percorso *“cd /usr/local/src/”* tramite il seguente comando:  git clone https://github.com/tesseract-ocr/tesseract  In seguito ho seguito i seguenti passi per completare la configurazione delle dipendenze:  cd /usr/local/src/tesseract  ./autogen.sh  ./configure  make  make install  cd /usr/local/src  wget https://github.com/Itseez/opencv/archive/2.4.13.zip  mv 2.4.13.zip OpenCV-2.4.13.zip  unzip -q OpenCV-2.4.13.zip  cd /usr/local/src/opencv-2.4.13  mkdir release  cd release  cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D BUILD\_NEW\_PYTHON\_SUPPORT=ON -D INSTALL\_C\_EXAMPLES=ON -D INSTALL\_PYTHON\_EXAMPLES=ON -D BUILD\_EXAMPLES=ON ..  Una volta fatto il comando *make* è partita un’installazione che è durata all’incirca una mezzoretta, nel frattempo ho continuato con l’implementazione del sito web.  Ho cominciato e terminato l’implementazione della pagina web che gestisce la visualizzazione dello storico, per completarla mi mancava solo l’implementazione di una funzione che contenga una query di estrapolazione dei dati necessari (nome, congome, n° targa, marca, colore, modello, tipo) e per fare ciò ho deciso di utilizzare una join.  Funzione getAllUsers()  /\*\*  \* Questa funzione si occupa dell'estrapolazione dei dati dalla tabella detentore e auto tramite una join.  \* @return list Array di stringhe contenente tutti i nomi, congnomi e le varie info della propria auto di ogni utente registrato.  \*/  function getAllUsers() {  $containerData = array();  //Stabilisco una nuova connessione con mysqli.  $this->newConnection();  //Query che estrapola il nome, il cognome e le info delle relative auto dei vari detentori.  //Utilizzo un prepare statement per evitare delle SQLInjection.  $stmtViewAllUsers = $this->conn->prepare("SELECT d.Nome, d.Cognome, a.N\_targa, a.Marca, a.Colore, a.Modello, a.Tipo  FROM detentore d, auto a  INNER JOIN auto  WHERE d.Email = a.Email\_detentore;  ");  $stmtViewAllUsers->execute();  $result = $stmtViewAllUsers->get\_result();  //Ciclo tutti i "dati" che la query mi ritorna e li inserisco in un'array.  while ($row = $result->fetch\_assoc()) {  array\_push($containerData, $row);  }  return $containerData;  }  Riempimento della tabella  <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">  <thead>  <tr>  <th>Nome</th>  <th>Cognome</th>  <th class="notRequired">N° Targa</th>  <th class="notRequired">Marca</th>  <th class="notRequired">Colore</th>  <th class="notRequired">Modello</th>  <th class="notRequired">Tipo</th>  </tr>  </thead>  <tbody>  <?php for ($i=0; $i < count($usersHistory); $i++): ?>  <tr>  <td><?php echo $usersHistory[$i]['Nome']; ?></td>  <td><?php echo $usersHistory[$i]['Cognome']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['N\_targa']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Marca']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Colore']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Modello']; ?></td>  <td class="notRequired"><?php echo $usersHistory[$i]['Tipo']; ?></td>  </tr>  <?php endfor; ?>  </tbody>  </table>  Test |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Non ho riscontrato problemi. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| In ritardo rispetto alla pianificazione. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Continuare con la configurazione del Raspberry PI |