

SCENARIO

While doing this project;

- 1- To ensure that new students who have come to Hacettepe University can find the buildings they are looking for on campus.
- 2- To ensure that they can quickly access information about that building.
- 3- To ensure that the accessed information can be viewed by everyone on the web intended.

ACCESSING THE HEROKU DATABASE

A database server should be available in the web to initiate a mobile geographic data collection project. Therefore, this tutorial relied on Heroku to set up the database on the cloud.
(<https://signup.heroku.com/>) You can open an account on heroku via this link.

1. Right click on Servers → Create Server
2. Click on General and provide a name for your server (e.g. heroku).
3. Go to the Connection tab, and enter the Heroku database credentials that you have previously obtained.
 - o Host name/address: Host
 - o Maintenance database: Database
 - o Username: User
 - o Password: Password (check the 'Save password' box for ease of login in future)

```
create table building_types
( name_ character varying(30),
  CONSTRAINT building_types_pkey PRIMARY KEY (name_));

create table buildings
( b_type character varying(30),
  id serial,
  b_name character varying(50),
  communication_no text,
  loc geometry('point',4326),
  observation_time timestamp,
  CONSTRAINT buildings_pkey PRIMARY KEY (id));

create table regions
( region_id serial,
  responsible character varying(30),
  region_geom geometry('polygon',4326),
  CONSTRAINT regions_pkey PRIMARY KEY (region_id));

create table admins
( admin_name character varying(30),
  CONSTRAINT admins_pkey PRIMARY KEY (admin_name));
```

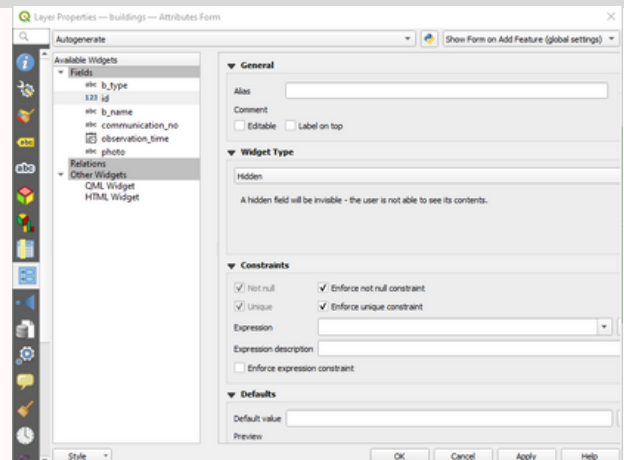
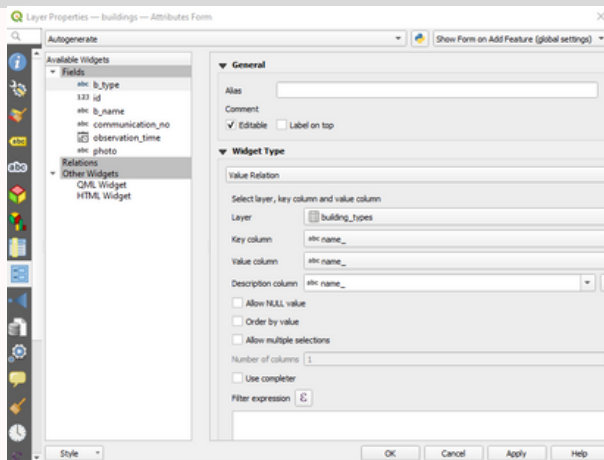
After connecting pgAdmin with Heroku, to open the query window, right click on the database and select the Query Tool. In this project, we need 4 tables: Buildings, Building types, regions and Admins tables for the administrators who will enter data. Then, using the insert into command, create the building type table to enter the building types, the building table for the buildings we will enter data into, the regions table to determine the regions for which the administrators are responsible, and the administrators table with the SQL statements on the side.

CONNECTING THE DATABASE WITH QGIS

The Postgres database created in the previous subsection can be included directly into a QGIS project, thanks to the seamless linkage between QGIS and Postgres. Once PostGIS is right-clicked, and New Connection is clicked, the following interface opens in which you can provide database credentials of the Heroku database.

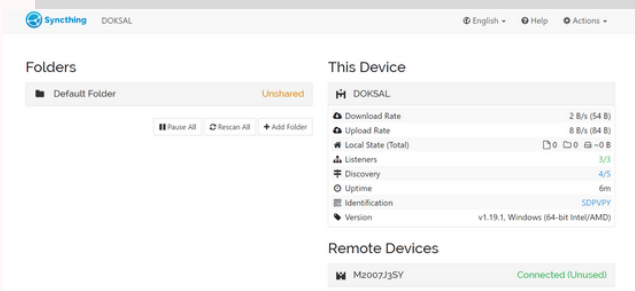
SETTING UP THE VALUE RELATIONS IN QGIS

In order to ease the process of data collection, it is useful to explicitly state the value relationships between the feature layers and to improve the user experience by hiding some of the details. For instance, a student can only select one of the pre-defined building types instead of typing it. This can be achieved by adjusting the Attribute Form specified under the Properties of the buildings layer as shown below. In addition, some of the attributes might better be hidden from the users such as the id attribute of the buildings as it is a serial number that auto increments as a building.



COLLECTING PHOTOS IN THE FIELD

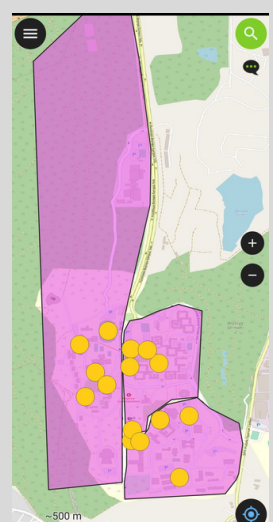
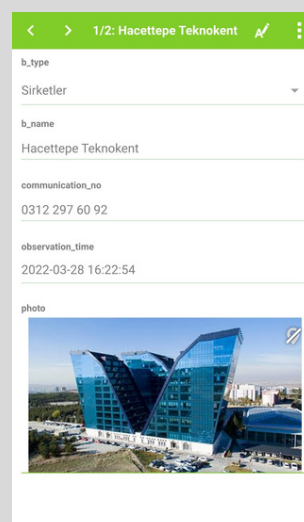
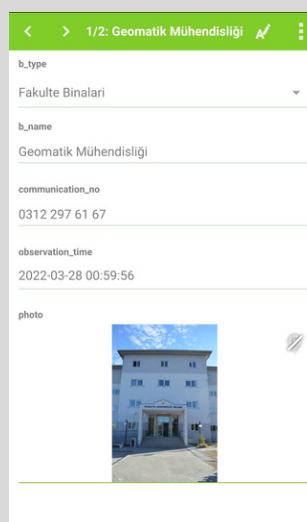
The "synthing-windows-amd64-" application was installed on the computer, and the "Synthing" application was downloaded on the phone. Then, the synchronization of the two devices was performed by entering the "application ID" on the computer to the application on the phone. Then after opening the attribute table of our "buildings" table, a column named "photo" was added and the data type was selected as "Text, unlimited length (text)". In the next step, the path to the file created on the computer and the phone was specified in the Widget Type section and our project was saved in this file with the name "osgeo_2022_w_photo".



Name	Date modified	Type	Size
DCIM	3/28/2022 1:17 AM	File folder	
osgeo_2022_w_photo	3/28/2022 1:53 AM	QGIS Project	71 KB
osgeo_2022_w_photo.qgs~	3/28/2022 1:30 AM	QGS~ File	67 KB

LOADING DATA VIA QFIELD

After the "osgeo_2022_w_photo" project was transferred to QFIELD, the option to add photos was also seen while entering data, and photos were added from the previously specified photo file on the phone.



WEB INTERFACE

For the web part;

First, an empty folder was created on the computer. After the created file is opened in Visual Studio Code and `npm init` is typed in the terminal, the metadata file `package.json` file is created.

Secondly, some javascript packages (`express`, `fs`, `bluebird`, `pg-promise`) were loaded by entering a command in the terminal.

In the next step, a file named "`appConfig.js`" is added to establish the database connection. "`appConfig.js`" contains database connection settings. In "`appConfig.js`" file Updated the "`connectionString`" part with the heroku link.

After that, the data stored in the database should be retrieved. A new `database.js` file was created for this. The link is implemented from `appConfig.js` and the query is implemented in this script.

Next, the website was created. A new `index.html` file was created. The background map is obtained from OpenLayers. Query response returned a GeoJSON file. So the JQuery library was used to get the query response.

Finally, the `index.js` file was created for the website to be published with NodeJS. This file provides the code to publish the website. "`https://gmt352qgis.herokuapp.com/`". When the browser was opened, the data stored in the database was retrieved and displayed on the map.

