Students of the
Geoinformatics Engineering
Masters Program

POLITECNICO
MILANO 1863

GEOINFORMATICS WEB APPLICATION

VERSION 1.2.

Wafi Mahdi Eldoud Mahdi
Person Code: 10694012/E-mail: wafimahdi@mail.polimi.it

8 Contents

1 Introduction:

1.1 Aim:

This report aims to showcase the technical details of the Geoinformatics students web application and describe all functionalities as well as the tools used for implementation.

1.2 Overview:

The web application is comprised of several pages that will allow users to view details about the students of Geoinformatics, which is represented in multiple methods and will also allow the users to login and share their own information.

2 System architecture:

2.1 Python:

2.1.1 Python scripts:

Main.py: handles the creation of the web application and all its functionalities.

Dashboard.py: handles the creation of the web application's data visualizations which includes a web map and several data charts.

The web application was developed using several python libraries each serves a specific function within the application. The libraries include:

2.1.2 Flask:

Flask is a web framework that provides tools, libraries and technologies that facilitate the creation of web applications. It is known as a microframework since it has little dependency on external libraries. Through Flask it is possible to create various functions that will make the base for the web application.

2.1.3 PyMySQL:

PyMySQL is an interface that establishes a connection between a python application and a MySQL database. Through this connection, it is possible to access all the tables within a connected database and be able to either retrieve data from the database or write data into it.

2.1.4 Flask Admin:

Flask admin is an administrative interface for Flask applications. It provides user-friendly tools for managing and interacting with data in a Flask-based web application. It allows the generation of database models, through which an administrator can create, read, update, and delete data from a given database.

2.1.5 Flask Mail:

Flask mail is an extension of Flask framework that allows the sending of emails from within a Flask application. It provides a reliable and efficient solution for managing email communication within Flask applications.

2.1.6 Subprocess:

Subprocess is a module used to run new scripts which spawn new processes from another python script. Through this module, it is possible to initialize and run a separate python script from the user interface of the web application.

2.1.7 Pandas:

Pandas is an open-source Python library used for data manipulation and analysis. It provides data structures and functions that make it easier to work with structured data such as CSV files, Excel spreadsheets, SQL databases, and more. The core data structure in Pandas is the DataFrame, which is a two-dimensional table-like data structure like a spreadsheet. DataFrames are highly efficient and offer a vast range of operations, allowing users to select, filter, transform, and aggregate data with ease.

Pandas also excels in data visualization. It integrates well with other Python libraries like Plotly, allowing users to create visually appealing plots, histograms, scatter plots, and more to explore and communicate insights from their data.

2.1.8 Folium:

Folium is a Python library that provides a simple and intuitive way to create interactive maps and visualizations. It is built on top of the widely used mapping library Leaflet.js and allows users to generate interactive maps directly within their Python environment. With Folium, it is possible to create various types of maps, such scatter plots, heat maps and more.

Folium makes it easy to overlay data onto maps by providing methods to plot markers, lines, polygons, and other shapes on the map canvas. The user can customize the appearance of these elements with options for colors, icons, pop-ups, and tooltips, making it a powerful tool for visualizing geospatial data. It also supports exporting maps as HTML files, which can be easily shared and embedded in web applications or notebooks. This makes it convenient for presenting and sharing visualizations.

2.1.9 Plotly:

Plotly is a popular Python library that provides a flexible and interactive framework for creating data visualizations. It offers a wide range of chart types, including line plots, scatter plots, bar plots, histograms, 3D plots, and more. One of the key features of Plotly is its interactivity. Plots created with Plotly are interactive by default, enabling users to explore the data by zooming, panning, and hovering over data points to view additional information. One of its standout features is the ability to save plots as HTML files, offering a convenient way to share and display interactive visualizations across different platforms.

2.1.10 Collections:

The collections module in Python provides specialized container datatypes that are more powerful alternatives to the built-in data structures. One of the most used classes in this module is Counter, which is a specialized dictionary subclass.

The Counter class is particularly useful in scenarios where it is needed to perform frequency analysis, find the most common elements, or determine the occurrence of specific elements in a collection. It simplifies counting tasks and provides a convenient interface to work with frequency data.

2.1.11 Base64r:

Base64r is a module that provides functions for encoding binary data to printable ASCII characters and decoding such encodings back to binary data. Encoding prevents the data from getting corrupted when it is transferred or processed through a text-only system. It allows image data to be decoded from binary data to pixel format.

2.1.12 MySQL Database:

MySQL is an open-source database management system which provides efficient ways to store, manage and retrieve data. The interface used to communicate with MySQL database is MySQL Workbench, which is a tool that allows users to manage their databases. Through MySQL Workbench users can create a new

database, edit an existing database as well as manipulate the tables stored inside any database. The database server is running on a localhost, which makes the development and testing of web applications more convenient. Connection to the database can be established through Python libraries.

2.2 Web Development:

For the visualization of the web application, the technologies used are:

2.2.1 Java:

A popular language for web development, known for its platform independence. Java is used to provide a web map from which locations can be extracted.

2.2.2 HTML:

Hypertext Markup Language, HTML, is the standard markup language for creating and structuring webpages. Through HTML, it is possible to organize and present data such as text, maps, images, etc. on a webpage.

2.2.3 CSS:

Cascading Style Sheets, CSS, is a style sheet language which describes the appearance of webpages. It works simultaneously with HTML to define the structure, fonts, colors, and the layout of the webpages. Making it more appealing.

3 Users:

Users of the web applications will be separated into two groups:

Admin: Coordinator of the application. Tasks include:

- Approval of new members.
- Removal of members.
- Contacting all users.

Members: Refers to users who registered to the web application.

Viewers: users who aim to view the web application without participating.

4 Functions:

Includes functions that require direct input from the web application's users.

4.1 Registration (/register):

Users: Viewers.

Registration will be available for viewers who wish to participate in the web application. The function will provide an input form asking for a username, email, and password. Upon submission, if the data entered by the user does not already exist in the database, their data will be stored as a new entry in the database and the user will be redirected to a login page.

4.2 Login (/login):

Users: Members.

The login page will ask the user to input their username and password to login to the web application. If the data entered by the user exists in the users' database, the user will progress into the web application's home page and will be granted member functionalities. Else, the user will be informed that the data they provided does not match any in the database.

4.3 Input (/input):

Users: Members.

By navigating the web application, the member can be redirected to an input page where they can provide personal data that will be published on a map in the web applications public home page, a personal profile page and a public page showcasing all the members of the application.

The data which the member is required to provide includes:

- Name.
- Surname.
- Gender.
- Data of birth.
- Undergraduate degree and university.
- Nationality.
- Hometown (name of city of birth and its location on a provided input map).

Each member will be allowed a single entry in the database and will be blocked from submitting multiple entries.

Members will also be asked to provide a personal picture.

Delete (/delete):

4.4 Users: Members.

Allows members to delete their accounts and remove all the data they submitted to the web application.

4.5 Logout (/logout):

Users: Members.

A function available for members which allows them to end their logged in session and change their status from members back into viewers.

Viewers will not be permitted to access the data input page and will not have a profile page. They will, however, be permitted to view the web application's home page, which contains data visualization derived from members' entries as well as a page featuring all the members names and other information.

4.6 AdminView (/admin):

Users: Admin.

Provides the web application's coordinator with the ability to approve new members, suspend the approval of current members and delete members.

4.7 Contact members (/send_email):

Users: Admin.

Allows the coordinator to contact all approved members of the web application via email.

4.8 Notify New Members (/new_members):

Allows the coordinator to notify potential members of their approval to use the web application.

5 Data Functions:

Includes the functions used to fetch the data stored in the database and display them on their designated webpages. All these functions were created using the python library Flask.

5.1 Upload (/upload):

This function receives image files submitted by members, changes the filename according to the uploaders username and stores the image file in a designated database table.

5.2 Data (/data):

The data function accesses the members' input table in the database and fetches back information about the logged in member to the web application and publishes it on a profile page. It also accesses a local folder where members' photos are stored and displays them next to their information.

The information retrieved by the data functions includes:

- Name.
- Surname.
- Gender.
- Data of birth.
- Undergraduate degree
- University.
- Nationality.
- Hometown.

5.3 Reload (/reload):

This function is used to run the Python script in charge of creating the members' data visualization figures. With every new entry from a new member, this function allows the web application to update the visualization methods with the newly entered data.

Students (/student):

This function accesses the members' input table in the database and fetches back information about all the members and displays it on a members' page. The information displayed includes:

- Name.
- Surname.
- Gender.
- Bachelor's Degree.
- University.
- Nationality.
- Position.

5.4 Non-members (/nonmembers):

Carries out the same process as the Students function, with the only difference being that it is accessible for non-logged-in users (viewers).

Render functions:

These functions are each assigned an HTML page to be rendered when called by the web application. They include:

- Index (/).
- Home (/home).
- Add (/add)
- Profile (/profile).
- Map (/map).
- Pie chart (/pie_chart).
- Bar chart (/bar_chart).
- Histogram (/histogram).
- Regions (/regions_chart).

6 HTML Templates:

The web application utilizes several HTML pages to display information as well as request it from members/viewers. They include:

6.1 Index (index.html):

Landing page of the web application. Through the index page, viewers can see the application's home and they can navigate to the functions that will allow them to become members.

6.2 Home (home.html):

This is the web application's home page. Can only be viewed by members.

Contents:

- An image slider which contains information about the Geoinformatics Engineering Master of Science program.
- A web map That contains members as markers, which can be expanded to view member details.

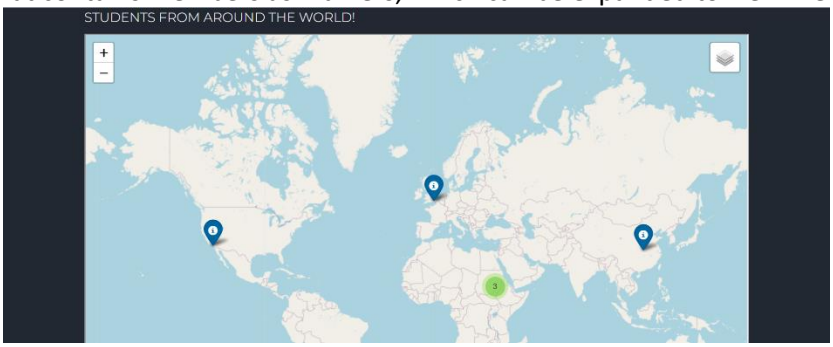


Fig (1): Students web map.

- A pie chart that showcases the members' gender distribution.
- Two bar charts the breakdown members by global region and nationalities.
- A histogram that details members' age groups.

Fig (2): member visualization charts.

6.3 Login/Registration (login.html):

Contains input forms That request either login or register information depending on the user's status.

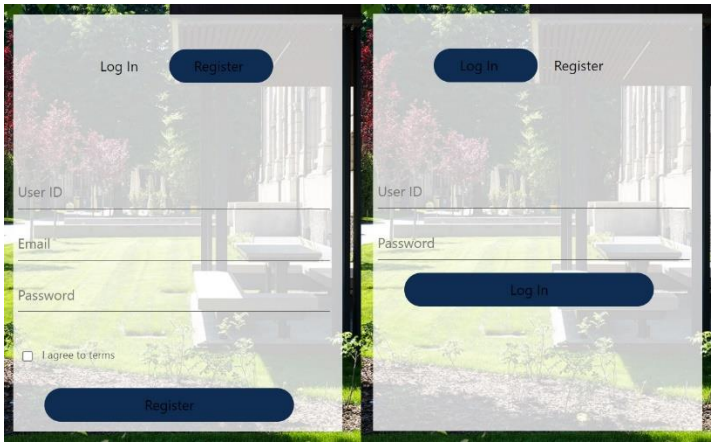


Fig (3): login and register forms.

6.4 Input (user_input.html):

Contains input forms That request personal information from members.

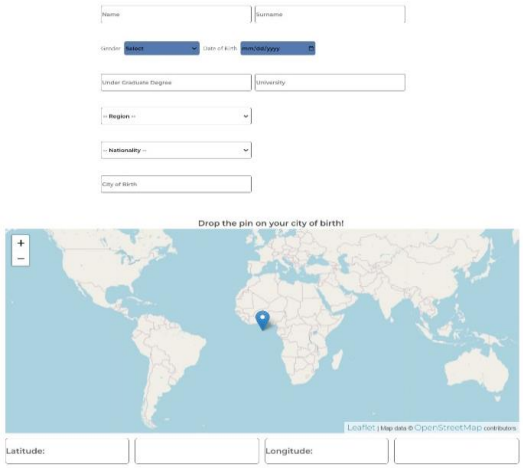


Fig (4): Member input form.

6.5 Upload (upload.html):

Ask members to upload their photo.

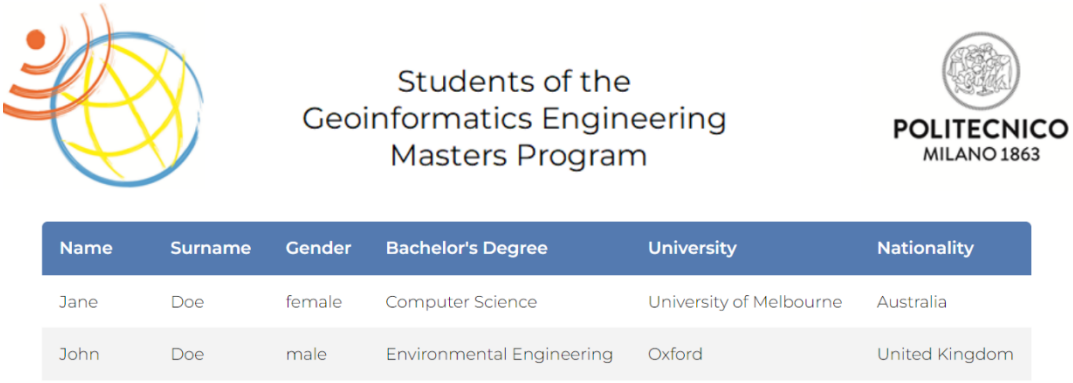
6.6 Profile (profile.html):

Contains some of the information submitted by the logged in member as well as their profile image. If the member did not submit any information, it displays a message stating that there is no data available.

Fig (5): Member's profile.

6.7 Students (student_data.html):

Features a table that contains the submitted information of all members at once.



Name	Surname	Gender	Bachelor's Degree	University	Nationality
Jane	Doe	female	Computer Science	University of Melbourne	Australia
John	Doe	male	Environmental Engineering	Oxford	United Kingdom

Fig (6): Members' data.

6.8 Admin Dashboard (admin/index.html):

The home to the coordinator's page. Includes paths to all admin functions.

6.9 All Users:

Auto generated template that views all users of the web application (approved and unapproved). Through this page the coordinator is able to change the approval status of members and delete members all together.

6.10 Contact Members (admin/mail.html):

Features a communication field through which the coordinator can contact members.

6.11 Notify New members (admin/mail_2.html)

Features a communication field through which the coordinator can notify new members of the change in their approval status.

6.12 Non-Members (nonmembers.html):

The same as the Students page, but accessible to viewers.

6.13 Students Map (output.html):

Automatically generated by the Python script (dashboard.py) and contains the students' web map.

6.14 Gender Pie Chart (pie_chart.html):

Automatically generated by the Python script (dashboard.py) and contains the students' gender pie chart.

6.15 Nationalists Bar Chart (bar_chart.html):

Automatically generated by the Python script (dashboard.py) and contains the students' nationalities' bar chart.

6.16 Regions Bar Chart (regions_chart.html):

Automatically generated by the Python script (dashboard.py) and contains the students' regions chart.

6.17 Ages Histogram (histogram.html):

Automatically generated by the Python script (dashboard.py) and contains the students' age groups' histogram.

7 Database design:

The web application's database, named "sql7629154", is an SQL database managed by MySQL workbench, a database management software. It is hosted on a free database hosting service at (freesqldatabase.com) and contains two tables described as follows:

7.1 Users Table:

Named "user" and it stores the registration information submitted by new members.

Table fields:

- ID.
- Username.
- Email.
- Person code.
- Password.
- Approval status.

7.2 Students Data Table:

Named "student_data" and contains the information submitted by members through the input form.
Table fields:

- ID.
- User ID
- Name.
- Surname.
- Username.
- Person Code.
- Gender.
- Birth Date.
- Undergraduate degree.
- University.
- Region.
- Nationality.
- Hometown.
- Latitude (in reference to the hometown).
- Longitude (in reference to the hometown).

7.3 Uploaded Files:

Named "upload_files" and contains the images uploaded by members and their unique identifiers:

Table fields:

- ID.
- User ID.
- File contents
- File Name.
- File extension.
- Username.