4992Report1010283.pdf

Yazar nevzatcan.cirpiciog@bahcesehir.edu.tr
nevzatcan.cirpiciog@bahcesehir.edu.tr

Gönderim Tarihi: 05-Haz-2022 10:39PM (UTC+0300)
Gönderim Numarası: 1850895108
Dosya adı: 4992Report1010283.pdf (6.44M)
Kelime sayısı: 7160
Karakter sayısı: 37432

T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES CAPSTONE PROJECT PROPOSAL

CALCULATION OF GLOBAL HORIZONTAL IRRADIATION AND PLANE OF ARRAY IRRADIATION

Energy Systems Engineering
Ahmad Alhaffar
Vedat Kaan Ay

Software Engineering
Sarper Bal
1730078
sarper.bal@bahcesehir.edu.tr 05316298837
Nevzatcan Çırpıcıoğlu 1806538
nevzatcan.cirpiciog@bahcesehir.edu.tr 05451661407

Advisor: Assist. Prof. Dr.Hüseyin Günhan ÖZCAN
Faculty of Engineering and Natural Science Deparment of Energy Systems Engineering

Advisor: Assist. Prof. Dr. Betül ERDOĞDU ŞAKAR
Faculty of Engineering and Natural Science
DepartmentDeparment of Software Engineering

ISTANBUL, December 2021

STUDENT DECLARATION

- By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that
- they have given credit to and declared (by citation), any work that is not their own (e.g. parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.);
- they have not received unpermitted aid for the project design, construction, report, or presentation;
- they have not falsely assigned credit for work to another student in the group, and not take ycredit for work done b another student in the group.

Abstract

CALCULATION OF GLOBAL HORIZONTAL IRRADIATION AND PLANE OF ARRAY IRRADIATION

Ahmad Alhaffar

Vedat Kaan Ay

Sarper Bal

Nevzatcan Çırpıcıoğlu

Faculty of Engineering and Natural Sciences

Advisors: Assist. Prof. Dr. Hüseyin Günhan ÖZCAN

Assist. Prof. Betül ERDOĞDU ŞAKAR

December 2021

Key Words: Global Horizontal Irradiation, Plane of Array Irradiation, PV Panel, Solar System, Solar Energy, Ground Reflactance, Reflactance of PV Module, Sky Diffusion, Tilt Angle, Normal Angle

Table Of Contents

List of Tables

List of Figures

LIST OF ABBREVIATIONS

PV	Photovoltaic
GHI	Global Horizontal Irradiation
POA	Plane of Array

OVERVIEW

In today`s World “Energy Generation” concept has no meaning by itself. Firstly it needs to be fulfilled with “Efficiency” term. In order to reach this idea; resources must be utilized the most possible optimum way. This is a greater aim since, serving today`s World gain importance while not taking from tomorrow`s.

In this context; Project targets to help increasing the efficiency of PV systems which are in a serious development process day by day. In addition to this, Project offers practical, clear and mobile app which will reflect all of those to the large scale of PV System users.

Ahmad Alhaffar and Vedat Kaan Ay from the energy systems engineering department and Sarper Bal and Nevzatcan Çırpıcıoğlu from the software engineering department will be collaborating in the task of this Project making.

1.1 Identification of Need

In this Project we have to achieve sustainable development goal announced by United Nations in energy, by that we are aiming to decrease the cost and increase the efficincy value of the solar energy.

While conducting this Project; POA irradiation, GHI, weather data, location, tilt angle values are needed with a considered time interval in order to conduct comparison and approach precision. Solar energy and solar PV technology is promising and has good potentials for electricity generation. Solar PV technology is mainy use in power production where plane of array radiation is the effective environmental parameter on it.

1.2 Definition of the Problem

Nowadays PV system usage has been broaden therefore, this wide range of users is required to have some crucial instructions such as tilt angle of the PV modules. The tılt angle of the PV panels have not been taken into enough consideration by the owners. Since the tilt angle is one of the most significant effects on the efficiency of the PV solar panels electricity generation, application is planned be a solution for that common disregard. With this Project it is mainly aimed to offer optimal, mobile and custom- based solutions. Meaning that all type of scale PV system users are targeted to be guided to modify their system into the most possible efficient state in spec of their own conditions.

1.2.1 Functional Requirements

It does not require a complex functional system since the greatest challenge is to approach precision. However; end- users may prefer to install autonomous system which will integrate the tilt angle according to the data that Project will offer.

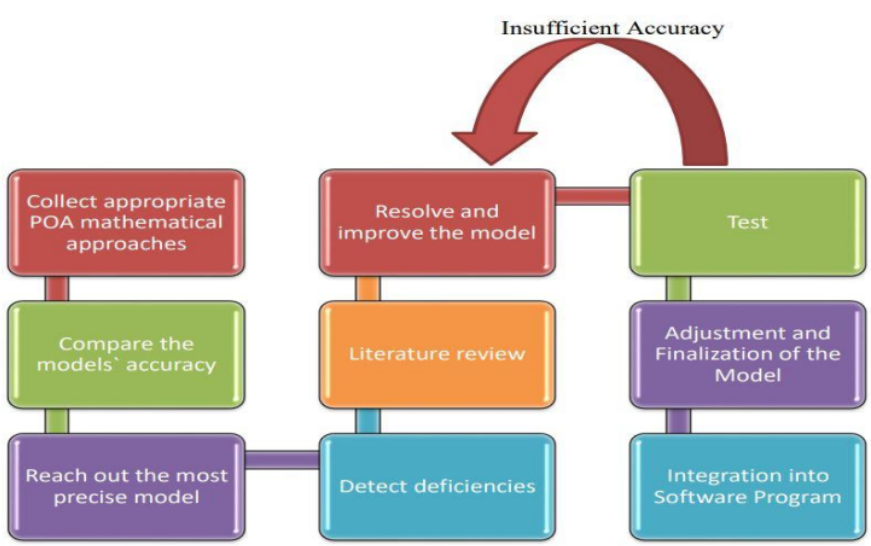
1.2.2 Performance Requirements

Precision is the key factor to evaluate qualification. Therefore, it is planned that the related proper mathematical models will be compared such as Liu Jordan, HDKR, Reindl, Badescu, Bugler. Results will be observed and the most precise one will be chosen to estimate POA related with the GHI. Gathered estimation will be closely compared with the real values which will be obtained from the companies or literature. If the final results are still very apart from each other, new development and approaches is going to be integrated into the model.

1.2.3 Constraints

It is aimed to reach the most appropriate actual results, in order to approach this; Experimental- based pathway should be followed. Such cost of carrying out multiple experiments is possible with some financial support. Otherwise past data tables will be used to improve functional accuracy.

1.4 Physical Architecture



2. Work Plan

Four main topics will be researched and derivated to create a custom equation:

- Global Irradiation:

Radiation value that is reaches to the earth in terms of $\frac{W}{m^2}$ unit

-Reflactance of PV Module:

The percentage of radiation reflected from the module. It is mostly related with the angle between the normal of the surface and the module glass type.

-Sky Diffusion:

Scattered radiation in the atmosphere which is prevented from reaching the surface of the earth.

-Ground Surface Reflectivity:

This is the factor that is being taken into account, where the radiation reflects from the ground surface. It does not have a significant effect compared with the other factors, however; some glittering and whitish colored surface may have distinguishable impact on POA.

2.2 Responsibility Matrix

Software Engineering Responsibility Matrix:

Task	Nevzat	Sarper
Planning	S	R
Front-End Development	R	S
Back-End Development	S	R
Database Development	R	S
Data Extraction and Crawling	S	R
Integration	S	R
Report	R	S

R= Responsible; S= Support

Table 1. Responsibility Matrix for the team for software engineering

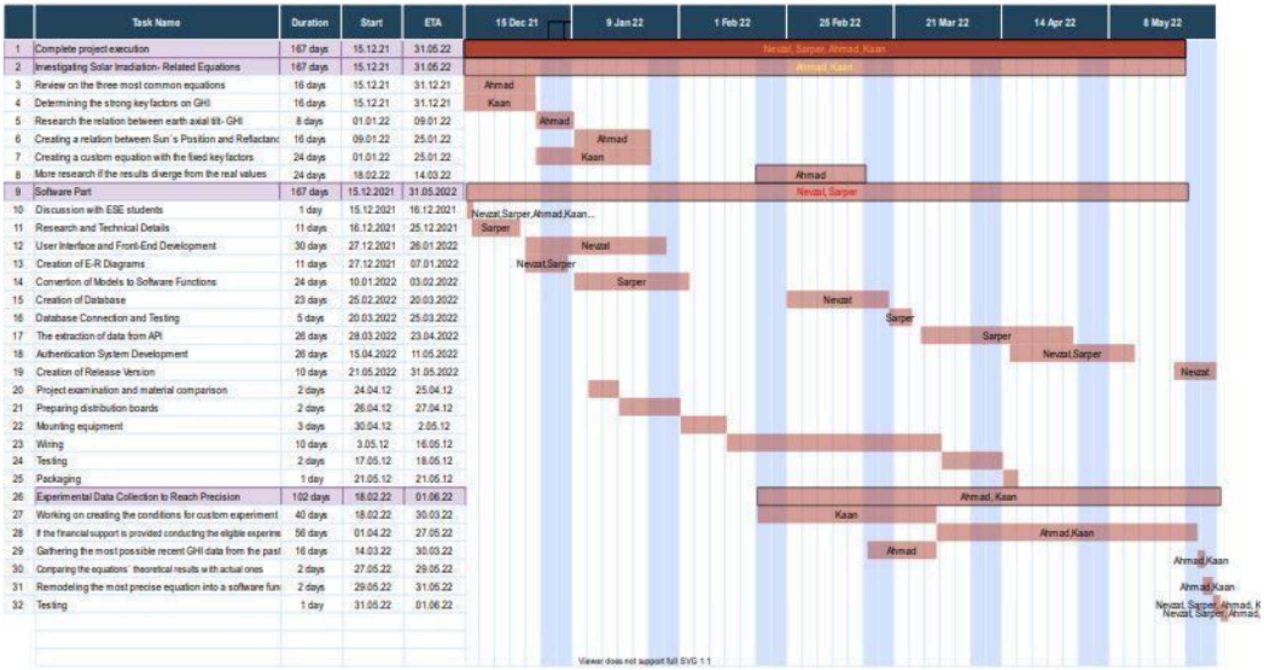
Energy Systems Engineering Responsibility Matrix:

Task	Ahmad	Kaan
Experimental data collection to reach precision	R	S
Investigating solar irradiation-related equations	S	R
Review on the three most common equations	R	S
Determining the strong key factors on GHI	S	R
Research the relation between earth axial tilt-GHI	S	R
Creating relation between Sun's position and reflectance	R	S
Creating a custom equation with the fixed key factors	R	S

R=Responsible; S= Support

Table 1.1 Responsibility Matrix for the team for energy systems engineering

2.4 Gantt Chart



	Task Name	Duration	Start	ETA
1	Complete project execution	167 days	15.12.21	31.05.22
2	Investigating Solar Irradiation- Related Equations	167 days	15.12.21	31.05.22
3	Review on the three most common equations	16 days	15.12.21	31.12.21
4	Determining the strong key factors on GHI	16 days	15.12.21	31.12.21
5	Research the relation between earth axial tilt- GHI	8 days	01.01.22	09.01.22
6	Creating a relation between Sun´s Position and Reflectanc	16 days	09.01.22	25.01.22
7	Creating a custom equation with the fixed key factors	24 days	01.01.22	25.01.22
8	More research if the results diverge from the real values	24 days	18.02.22	14.03.22
9	Software Part	167 days	15.12.2021	31.05.2022
10	Discussion with ESE students	1 day	15.12.2021	16.12.2021
11	Research and Technical Details	11 days	16.12.2021	25.12.2021
12	User Interface and Front-End Development	30 days	27.12.2021	26.01.2022
13	Creation of E-R Diagrams	11 days	27.12.2021	07.01.2022
14	Convertion of Models to Software Functions	24 days	10.01.2022	03.02.2022
15	Creation of Database	23 days	25.02.2022	20.03.2022
16	Database Connection and Testing	5 days	20.03.2022	25.03.2022
17	The extraction of data from API	26 days	28.03.2022	23.04.2022
18	Authentication System Development	26 days	15.04.2022	11.05.2022
19	Creation of Release Version	10 days	21.05.2022	31.05.2022
20	Project examination and material comparison	2 days	24.04.12	25.04.12
21	Preparing distribution boards	2 days	26.04.12	27.04.12
22	Mounting equipment	3 days	30.04.12	2.05.12
23	Wiring	10 days	3.05.12	16.05.12
24	Testing	2 days	17.05.12	18.05.12
25	Packaging	1 day	21.05.12	21.05.12
26	Experimental Data Collection to Reach Precision	102 days	18.02.22	01.06.22
27	Working on creating the conditions for custom experiment	40 days	18.02.22	30.03.22
28	If the financial support is provided conducting the eligible experime	56 days	01.04.22	27.05.22
29	Gathering the most possible recent GHI data from the past	16 days	14.03.22	30.03.22
30	Comparing the equations´ theoretical results with actual ones	2 days	27.05.22	29.05.22
31	Remodeling the most precise equation into a software fun	2 days	29.05.22	31.05.22
32	Testing	1 day	31.05.22	01.06.22

Table 2. Gantt chart for the complete project execution

2.6 Risk Assessment

RISK LEVEL		**Severity** of the event on the project success		
		Minor	Moderate	Major
Probability of the event occuring	Unlikely	VERY LOW	LOW	MEDIUM
	Possible	LOW	MEDIUM	HIGH
	Likely	MEDIUM	HIGH	VERY HIGH

VERY LOW	This event is very low risk and so does not require any plan for mitigation. In the unlikely event that it does occur there will be only a minor effect on the project.
LOW	This event is low-risk; a preliminary study on a plan of action to recover from the event can be performed and noted.
MEDIUM	This event presents a signficant risk; a plan of action to recover from it should be made and resources sourced in advance.
HIGH	This event presents a very signficant risk. Consider changing the product design/project plan to reduce the risk; else a plan of action for recovery should be made and resources sourced in advance.
VERY HIGH	**This is an unacceptable risk. The product design/project plan must be changed to reduce the risk to an acceptable level.**

Table 3. Risk matrix

Failure Event	Probability	Severity	Risk Level	Plan of Action
Server Failure	Likely Server providers usually are reliable	Major	VERY HIGH	Keeping existing data safe. Temporarily inaccessible system.
External Database Connection Error	Possible	Major	HIGH	The error "Unable to access information source temporarily" is displayed.
Local Database Connection Error	Unlikely The server is largely safe in this regard.	Moderate	LOW	Database backup is made.
Computation Error	Possible	Minor	LOW	Improving the mathematical model

Table 4. Risk assessment

SUB-SYSTEMS (SOFTWARE PART)

3.1 The Name of the Subsystem 1

We compiled the software side of the project in the 3rd part (sub-system 1) of the report. This section has been prepared by the software engineering team. The structural and administrative aspects of the software project are aimed to be explained and visualized here.

3.1.1 Requirements

Behaviors of the Software Application

Actor Name	Name of the Behaivor	Description of the Behaivor
User	Login Sign up Send Form Data View	It is the user who uses the system, can log in to the system with e-mail and password, has the authority to change the password, and can view data by entering coordinate information into the system.
Technical Support	Support	The person who tries to solve the problem by providing technical support when any system-related error occurs
Compute Service	Data Processing Computing Functions	According to the coordinate information entered by the user, the data retrieved from the web is calculated through mathematical functions.
Database System	Queries Storage Data	User e-mail addresses, passwords, and payment information are registered in the database.

Authentication Service	Authentication	The service that checks whether the email and password entered by the user are registered to the system.
Payment System	Monthly or annually payment	It is the system that enables payment transactions between users and the software system.
Control Service	Control User Information	It is the system that checks whether the number of characters in text boxes used for passwords is more than 10 letters and valid email addresses.
E-Mail Verification System	Two-factor Authentication	The system checks the existence of the e-mail address and the consistency of the passwords entered by the user.

Attributes of the Software

Actor Name	Name of Attribute	Description of Attribute
User	Email Password User ID Payment Information	Email: It is the e-mail address and password used by the user to log in to the system. User ID: The unique ID is given by the system after the user registers in the system. Payment Information: Information indicating whether the payment has been made.
Technical Support	Support ID Email Password	Support ID: ID number that identifies the user as support. Email and Password: It is the e-mail address and password used by the technician to log in to the system.
Compute Service	Math Functions Location Information Data	Math Functions: Functions that calculates data Location Information: Location that is entered by the user to the system to view that location's data. Data: Retrieved data from the web API to calculate.
Database System	Table	Table: Tables in which user information is kept according to the appropriate data type.

Authentication Service	Email Password	Email and Password equal user email address and password.
Payment Service	Payment Approval Info	Payment confirmation information
Control Service	Email Password	Email and password validation

Non Functional Requirements

1- The system will be accessible 24/7.
2- The system will give a warning message when the email or password is incorrect.
3- User password must be at least 1 uppercase letter, at least 1 digit, and must be at least 10 characters.
4- According to the location, the data should be shown to the user within 10 minutes.
5- When the user breaks any system-related rules, the system shall give a warning message.
6- User interfaces shall be easy to use on the web or desktop.
7- The system shall hide users' personal information such as passwords.

Functional Requirements

1. User shall enter the system.
2. Users shall change their passwords.
3. Users shall be able to reset their password when they forget it.
4. All users shall contact technicians.
5. Users shall select location information.
6. The system shall send notifications about information for the process.
7. E-mail service shall send an email which context is "your password has been changed" when the password is changed.
8. Users shall view locations.
9. Engineers shall be allowed by the system for organizing all functions.
10. All users shall communicate with support.
11. The system shall calculate data from the API.
12. The user shall be able to view the calculated data.

Requirement	Importance
Keeping user data safe	High
Obtaining the necessary information from the user and saving them in the database.	High
Delivery of correct information to users.	High
Creating independent development layers for easy development.	High
The relational and non-relational database is designed in a clear and secure way.	High
Creating screens that users can easily use and understand.	Medium
Designing the HTML DOM structure to be stable and to spend low performance. For good user experience	Medium
To make beautiful and attractive page designs.	Low

3.1.2 Technologies and Methods

Basically, we planned our technical structure as a 3-layered architecture. These layers have the names Presentation (Front-end, Interface), Processing (Back-end, Logic) and Data. The reason we implement this architecture is to improve the application security and development process. The server-side development process can be done independently of the development of the web and desktop application, thus making more good task distribution, and reducing interdependencies. The gains for us in the field of security will be that the error in each layer will affect other regions less. Abstraction between layers has been one of the important points in our architectural choice. We avoided monolithic application architectures as much as possible and created a more distributed architecture.

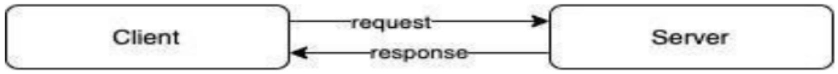
Presentation/ Front-end/ Interface Layer

In this layer, we created the visual and user experience of our application. Basically, we designed our pages using HTML, CSS, and JavaScript. It is the layer where we aim to create a UX (user experience) for the user with the UI (user interface) offered to the user. Thanks to this layer, the user communicates with the server and uses the functions she needs.

UI (User İnterface): UI stands for user interface, and it is aimed for experience- oriented improvements such as users to navigate the site comfortably and stay on the site for a longer period and access the site content more easily. In our application, we aimed to design a simple and useful interface for the user to easily enter location information and visualize the results as desired.

UX (User Experience): We aim to create a good user experience with a simple interface design and a server structure that responds as quickly as possible.

Note: This planning is done assuming a web application will be developed.



Server: It is the party requested by the client, that is, the party that runs the requested pages.
Client: It is the party that requests the pages from the server and displays them to the user. This client is a web-server.

Client-Side Programming is the coding style that the written codes work on the browser on the client-side. These codes do not work on the server, they are interpreted by the browser. In web applications, these are browsers, codes that run on the user's computer. Usually, JavaScript, HTML, CSS are used.

Advantages: The server is not overloaded, the codes run on each client's computer.

Disadvantages: Performing all calculations here may cause problems in terms of data security. Browsers may not always show the same performance and the results may have errors.

We will perform the visualization processes of the data coming from our application in the client-side section. By using JavaScript libraries, we will enable users to visualize the data they want, to give an example, the Chart.js library can be given.

Processing (Back-end, Logic) Layer

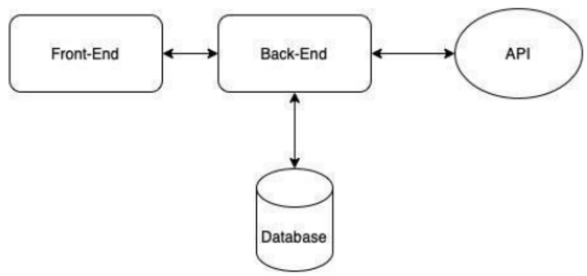
This layer is basically processing the information it receives from the interface layer and sending it to the database. In case of request, information can be transmitted to users in the database. It is the basic layer where the application performs its functions and meets the requirements. It is an important layer for security as it controls access to the database.

Server-Side Development

Server-side programming is a general name given to all applications running on the server. Its main job is to dynamically generate content and send what the client wants.

Advantages: Since the source codes are not sent from the server to the client, they cannot be seen by anyone. Its access is secure against unwanted persons. Since the source codes are run on the server-side, the dependency on the browser is eliminated.

Disadvantages: Since all the codes are running on the server-side, there is a lot of loads on the server. Costs can be very high.



API (Application Programming Interface): Thanks to the APIs here, we will be able to instantly pull the data we need from 3rd party applications and apply the operations we want in the back end.

In the back-end section, we will develop our application in a cloud service. There are 3 major service providers here. These are Google Cloud, AWS, Microsoft Azure. We will use the services of these applications while creating our cloud architecture.

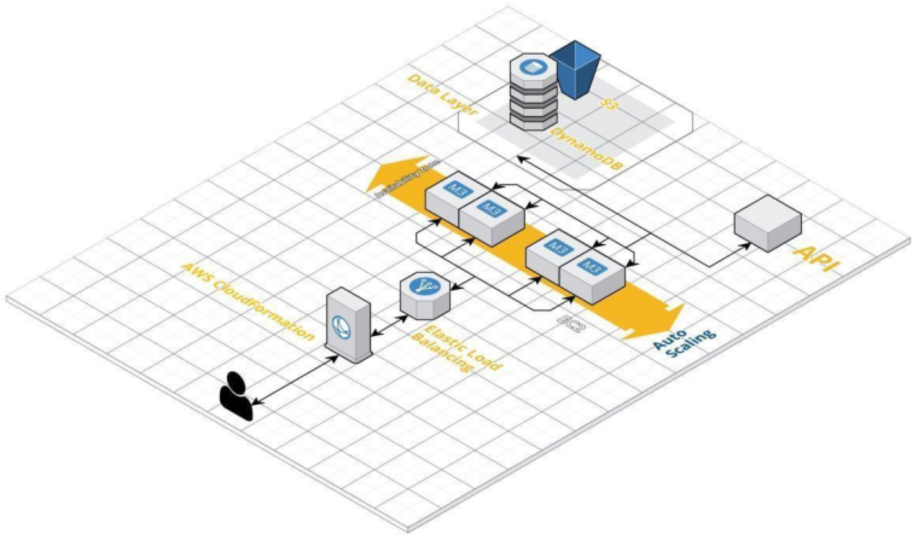
Note: In the next examples, I will create the schemas using AWS services, but there are equivalent services on other cloud servers. Infrastructure is portable to other cloud services.

Cloud Architectures

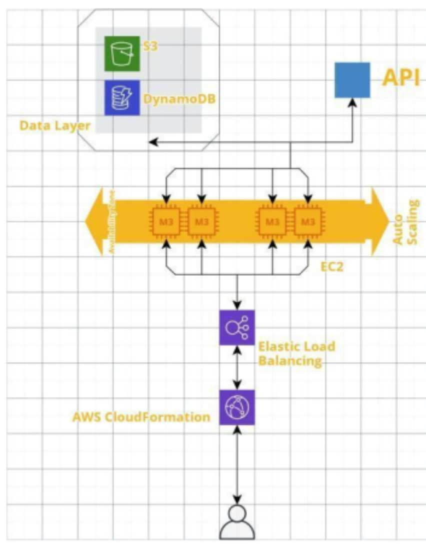
IaaS

It contains the basic building blocks for the IT system and generally provides access to network features, computers, and data storage. We can think of it as a cloud service where virtual Compute, Storage, Networking are sold or leased over the cloud. Instead of buying computers, disks, and network cards, we can rent them from the cloud virtually and install the desired operating system or applications on them and create solutions.

-3D diagram



-2D diagram



AWS Cloud Formation: It is a service that runs the application on the nearest server and offers the best user service, according to the region the users enter.

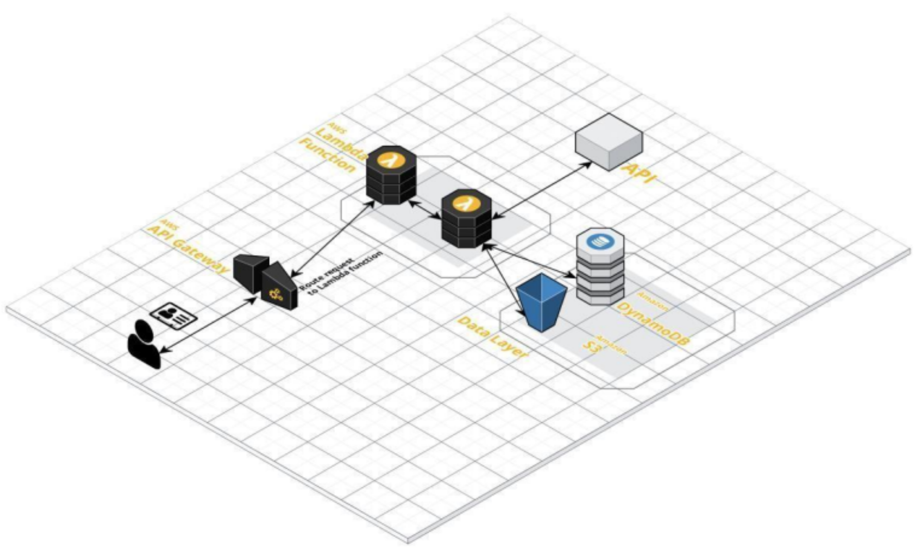
AWS EC2: It is the virtualization service of AWS. It is the service that opens machines with the features you want and allows you to install applications. The mathematical operations of our application will be performed on these machines.

AWS Elastic Load Balancing: It is a service that allows scaling EC2 machines. New machines can be turned on or off according to incoming requests. For example, if there is a large workload, this service detects this and enables new virtual machines to be opened.

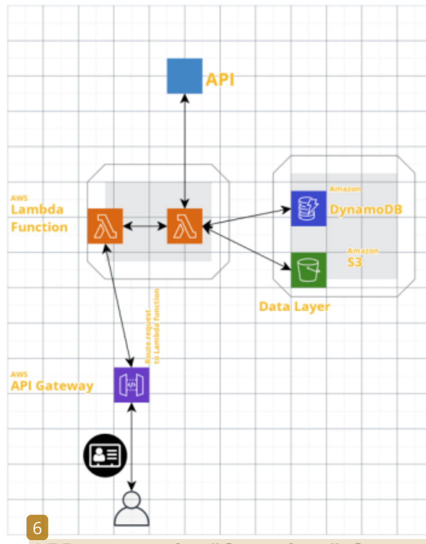
FaaS and Serverless

With FaaS and Serverless architectures, it will be much easier to develop, scale and maintain software. At least, the claims of cloud platforms, especially AWS, are in this direction. Many examples such as AWS Lambda, Amplify, Cloud Functions and Firebase exist today, and significant investments are being made in this regard. In short, FaaS is just the builds where you load your functions and want it run for you. Being able to scale automatically, paying as you use it, and having a very agile structure are the biggest advantages they offer you.

-3D diagram



-2D diagram



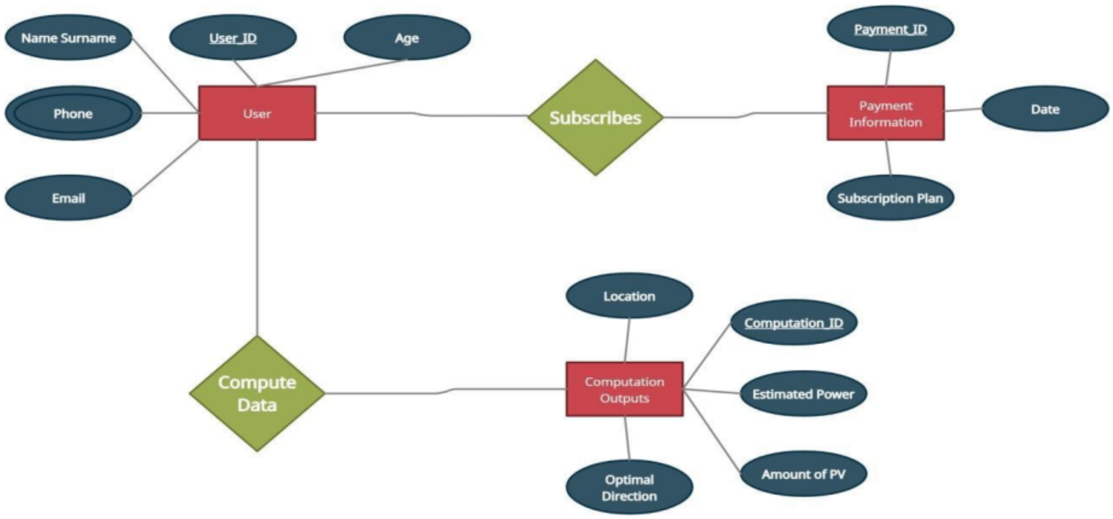
AWS API Gateway: APIs act as the "front door" for applications to access data, business logic or functionality from your backend services. API Gateway is a port that transmits the data produced with Lambda in our project to the user via API and transmits the information from it to the Lambda service.

AWS Lambda: AWS Lambda is a serverless, event-driven compute service that lets you run code for virtually any application or backend service without provisioning or managing servers. In our application, a series of Lambda services have been designed to retrieve data from the API using the location information from the user and transmit it to the end-user and the database after processing it.

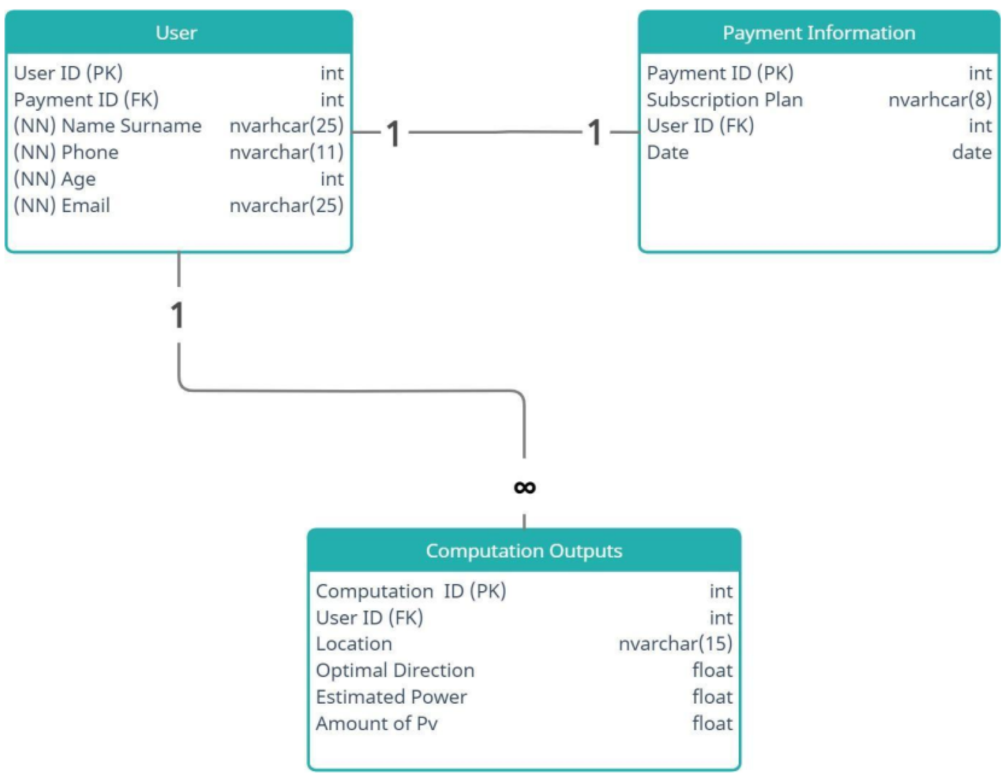
Data Layer

In this layer, we host our basic data and their basic relationships with each other. This data can be queried or changed by requests from the logic layer. Application data is kept in different database types such as relational database, non-relational database, and object storage. Database types can be decided according to the needs that occur while the application is being developed.

E-R Diagram



Database Tables



3.1.3 Conceptualization

Use Case Glossary

Use Case Name	Description	Participating Actors
Sign up	This use case describes that users register to the system.	• Users • Control Service
Login	This use case describes logging into the system.	• Users • Database System • Authentication System
Reset Password	This use case describes allowing users to set a new password.	• Users • Database System • Authentication System • Control Service
Choose Payment Plan	This use case describes allowing users to choose one of the payment plans	• Users • Authentication System • Database System • Payment System
Compute Solar Radiation Data	This use case describes that the data which is retrieved by API is calculated.	• Users • Compute Service • Database System
Graphically Show Data	This use case describes allowing users to view the data which is calculated by the compute service graphically.	• Compute Service

Use Case Scenarios

Use Case Name:	Sign up
Use Case Description:	Users must be registered to use the system. To register, users must continue with "sign up". In this part, the system asks for name, surname, age, and email address. Then the user sets a password. After the registration process is completed, the user is directed to the "Payment Plan" screen.
Primary Actor(s):	User clicks the sign-up button on the system's user interface to register to the system.
Supporting Actor(s):	The control service checks whether the information entered by the users is compatible with the desired criteria.
Triggers:	Users who want to use the system register.
Preconditions:	There is no precondition for sign-up
Postconditions:	The system will redirect the user to the "Payment Plan" screen.
Normal Flow:	● Users who want to register will continue with sign-up. ● User will enter name, surname, age, and e-mail address. ● The control service will check whether the entered information is appropriate. ● Users will be redirected to the "Payment Plan" screen.

Alternate Flows:	● If the user does not set a password in the desired format, an error message appears on the screen. ● If the user does not fill in all the required information, the registration will not be made.

Use Case Name:	Login
Use Case Description:	Users must log in to use the system. Users log in to the system by entering their registered e-mail address or id numbers and password.
Primary Actor(s):	When the system is turned on, the user logs into the system by entering the id number or e-mail address and password.
Supporting Actor(s):	The authentication service checks whether the e-mail address or id number exists in the system and the password is compatible.
Triggers:	Users who want to use the system login.
Preconditions:	The user must be registered.
Postconditions:	System functions become available.
Normal Flow:	User id or e-mail address and password are entered and the system is logged in.
Alternate Flows:	• If the user id, e-mail address, or password is entered incorrectly, an incorrect login message will appear on the screen. • If the user is not registered in the system, the user must be registered. • If the user forgot her/his password, the user continues with the forgot password section.

Use Case Name:	Reset Password
Use Case Description:	Users who forget their password must reset their password. Users enter their email addresses to reset their passwords. Then a link will be sent to the email address so that the user can reset their password. Through this connection, the user sets a new password for himself/herself
Primary Actor(s):	User clicks the reset password button to reset when forgot his/her password.
Supporting Actor(s):	• The control service checks whether the information entered by the users is compatible with the desired criteria. • The authentication service checks whether the e-mail address or id number exists in the system. • The password is updated in the database.
Triggers:	Users who forget the password.
Preconditions:	There is no precondition to reset the password.
Postconditions:	Password will be changed and updated in the database. User will be redirected to the login page.
Normal Flow:	• The registered e-mail address is entered into the system. • A new password is determined with the verification link. • The user is directed to the login screen.
Alternate Flows:	• If the user id, e-mail address, is entered incorrectly, an “incorrect email address” message will appear on the screen.

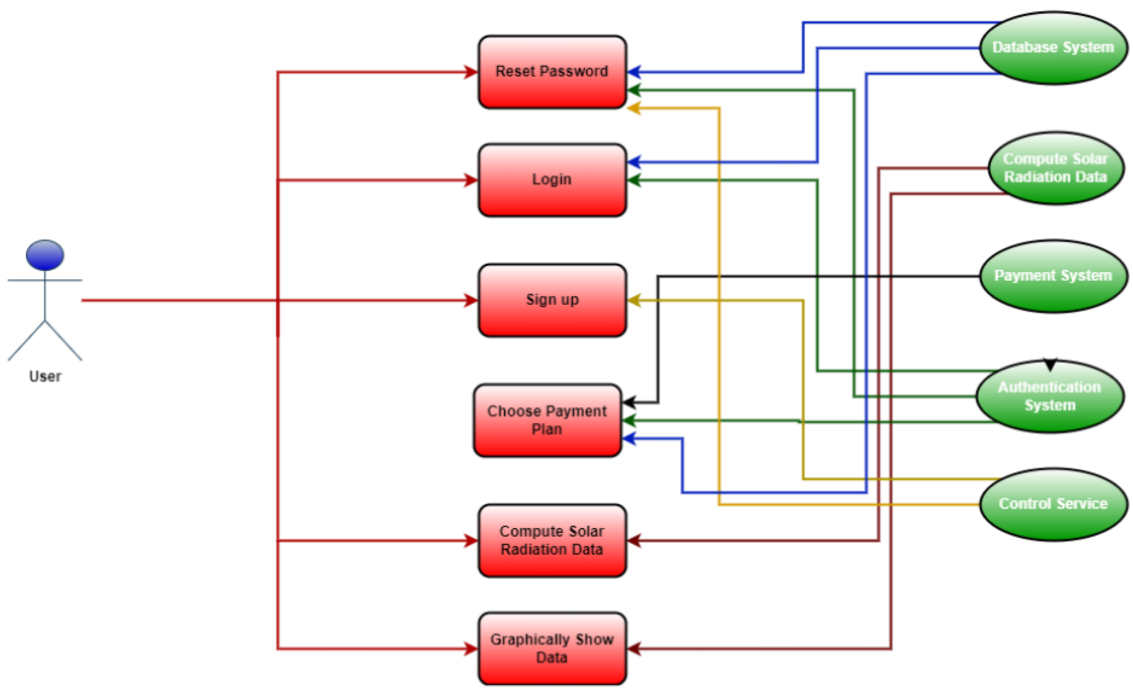
Use Case Name:	Choose Payment Plan
Use Case Description:	Users must choose a monthly or annual payment plan in order to register with the system. After the payment plan is selected, the necessary information for the payment process is entered. If the information is verified, the payment is made and the user is registered in the system.
Primary Actor(s):	The user displays the plans by clicking the select payment plan button and enters the necessary information.
Supporting Actor(s):	The payment system checks the accuracy of the information entered by the user and whether there is a sufficient balance for the payment transaction. If the balance is sufficient, the payment will be processed. User information is registered in the database.
Triggers:	Users who want to use the system need to select a payment plan and subscribe.
Preconditions:	User information must be entered (name, surname, age, email address, and password) and must be approved by the system.
Postconditions:	The payment process is completed. The user record is created.User id will be given.User information will be registered in the database.User will be redirected to the“Login Screen”.

Normal Flow:	● Card information (card no, security no etc.) is entered into the system. ● The information entered by the payment service and the balance are checked. ● The payment process is completed. ● User id is given. ● User information is recorded in the database. (not payment information) ● The user is directed to the Login screen.
Alternate Flows:	● If the card information is missing or entered incorrectly, the system will give an error message and the registration will not be performed. ● If the account balance is insufficient for the selected payment plan, the system will give an error message and the registration will not be performed.

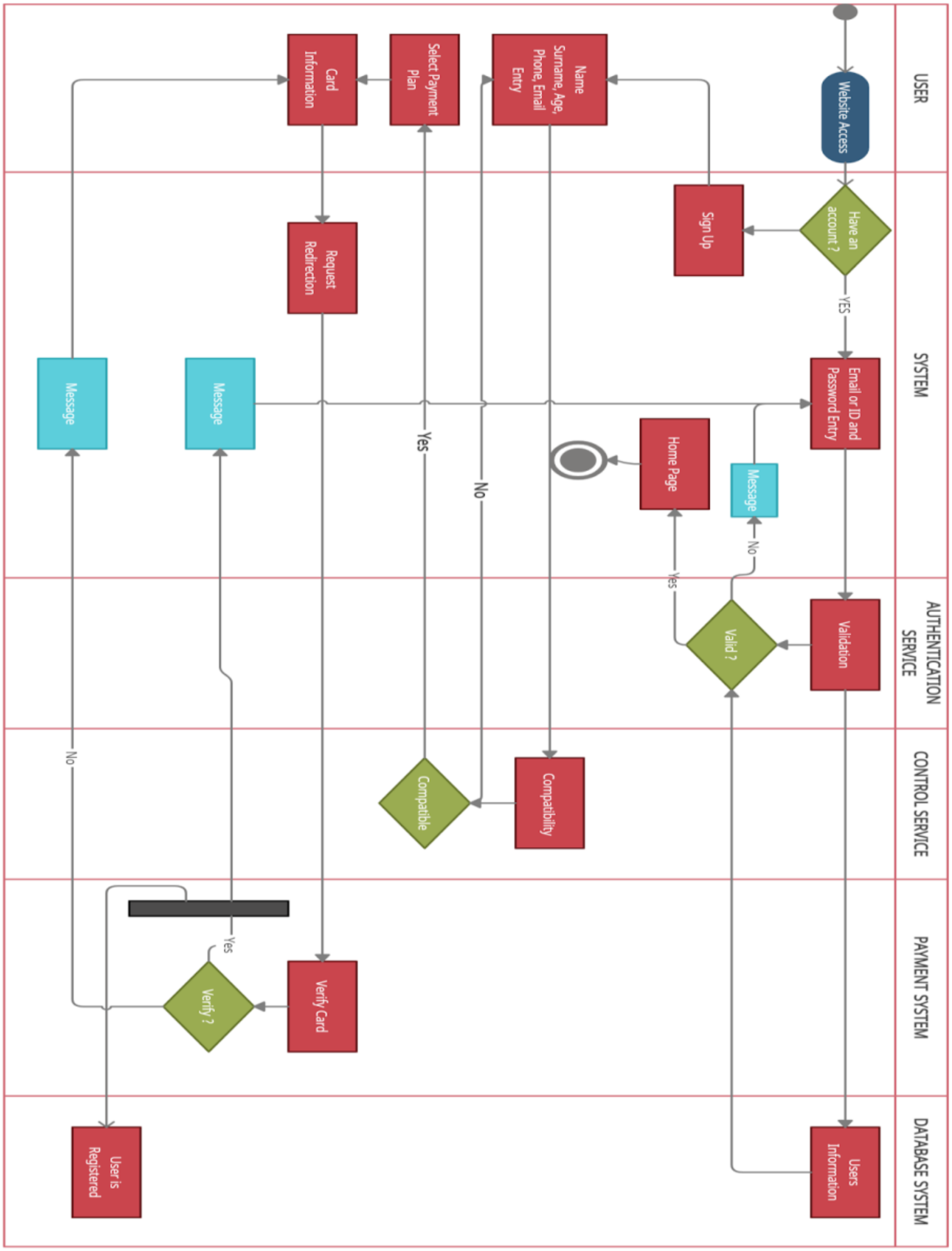
Use Case Name:	Compute Solar Radiation Data
Use Case Description:	The user will select a location through the system. According to the selected location, the data will be calculated through mathematical functions. The calculated data will be displayed to the user and saved on the database.
Primary Actor(s):	User will choose location
Supporting Actor(s):	The calculation service receives the data of the selected location and integrates it into mathematical functions. Displays the results of the function in the system.
Triggers:	User request
Preconditions:	User selects the location.
Postconditions:	Calculated data will be shown on the system.
Normal Flow:	● The user selects the location that he/she wants to be calculated. ● The data of the selected location is retrieved. ● The captured data is calculated in functions. ● The calculation result is shown to the user.
Alternate Flows:	If instant data of the selected location cannot be found, the calculation is not performed.

Use Case Name:	Graphically Show Data
Use Case Description:	After the calculation of the data, some outputs are obtained. The user can view these outputs graphically and download them to his/her device.
Primary Actor(s):	User clicks “Show Graphically” button.
Supporting Actor(s):	Compute service graphically displays the calculated data.
Triggers:	User Request
Preconditions:	Functions need to be calculated.
Postconditions:	Calculated data is displayed graphically.
Normal Flow:	When the user wants to see the data graphically, she/he views it with the "show graphically" option.
Alternate Flows:	If the functions are not calculated, the user cannot see the results graphically.

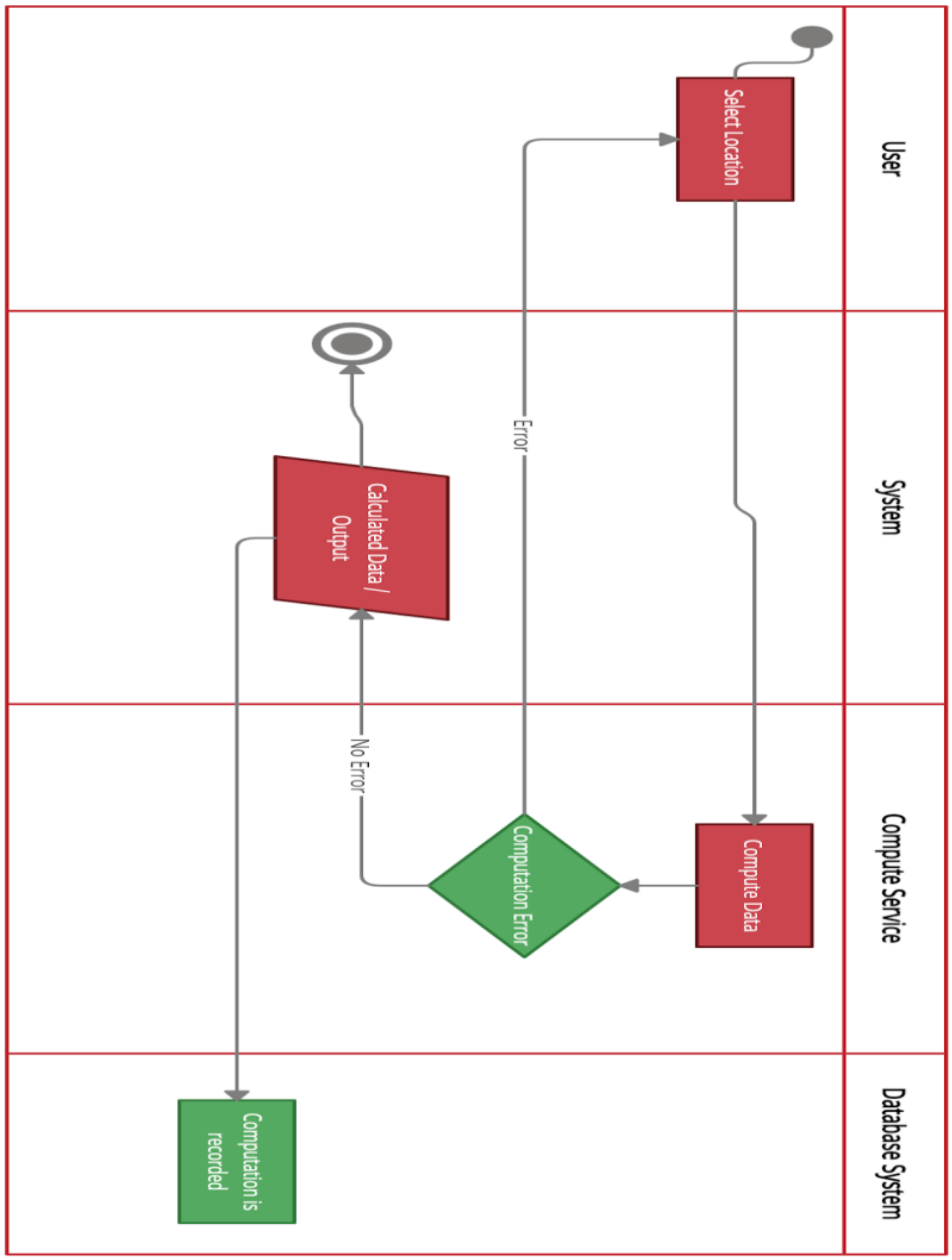
Use Case Diagram



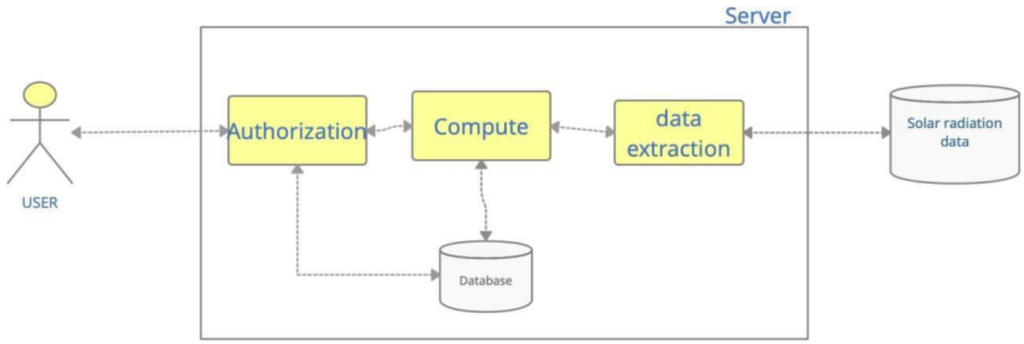
Activity Diagram – Authentication and Authorization



Activity Diagram Computation of Data



3.1.4 Physical Architecture



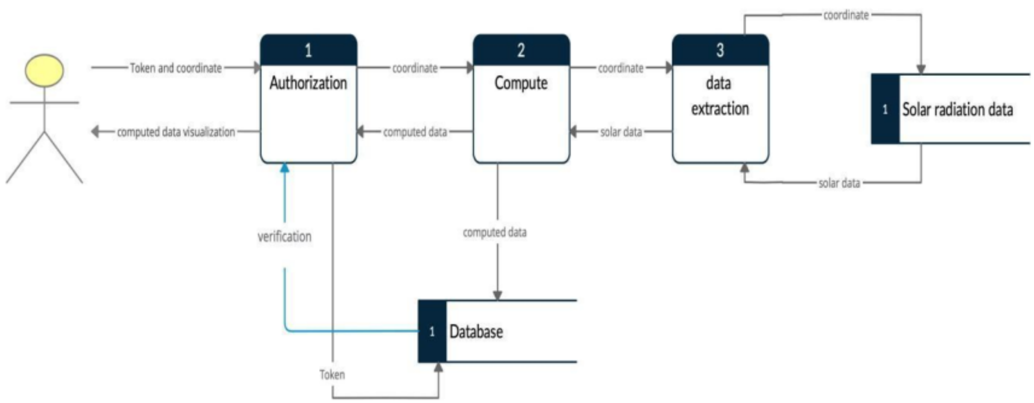
Solar Radiation Data: Data, that will be used to calculate mathematical functions, will be retrieved from websites.

Data Extraction: By sending a request to the API, the required data will be retrieved from third- party applications.

Compute: In order for the data coming from the websites to be calculated in the functions, conversions are made to the appropriate data type. Then the function results are saved in the database.

Authorization: The user logs into the system with her/his e-mail and password.

User: The user who sees the calculated data.



User information is verified by the system in the data set. User information is kept in the database. If the verification is successful, the user enters the coordinate information on the system. The data of the entered coordinate information is retrieved from the website. The data is converted into appropriate data types. The transformed data is calculated in the functions. Function outputs are saved in the database and then displayed to the user.

3.1.5 Materialization

POA models:

- PV performance models use inputs of plane of array (POA) irradiance and module temperatures in conjunction with details on the system configuration to estimate the DC and AC output of the PV system. Ideally, onsite measurements of POA irradiance, module temperature and maximum power of the PV array under operating conditions would be available as input into the PV performance models.

-POA irradiance models are typically dependent on inputs of global horizontal, diffuse and direct normal irradiance whilst the module temperature models are dependent on the parameters of POA irradiance and ambient temperature.

$$POA = GDirect + GDiffuse + GReflected$$

$$GDirect = DNI * cos(AOI)$$

$$GDiffuse = DHI * \frac{1 + \cos(tilt)}{2}$$

$$GReflected = GHI * Albedo * \frac{1 - cos(tilt)}{2}$$

- *GReflected* is often referred to as *RHI*, reflected horizontal irradiance,

2. Models tested:

1- Hay model (Hay-Davies model):

Provides predictions of the PV energy and the module temperature with a difference up to 1%.

7

The diffuse solar radiation on a tilted plane with tilt angle β and azimuth γ as given by Hay's Model is:

$$I_{D,\beta} = I_D\left[F\frac{\cos(\theta)}{\cos(Z)} + (1-F)\cos^2\left(\frac{\beta}{2}\right)\right]$$

2- Skartveit and Olseth model:

Require hourly clearness index and solar elevation as input.

7

Skartveit and Olseth's model was developed for high latitudes and is essentially an extension of the Hay model:

$$I_{D,\beta} = I_D\left[F\frac{\cos(\theta)}{\cos(Z)} + B\cos(\beta) + (1-F-B)\cos^2\left(\frac{\beta}{2}\right)\right]$$

B = max(0.3 – 2F; 0.0)

3- Gueymard model:

Gueymard presented a physically based irradiance model for tilted surface which its components are derived from typical patterns and the results have good match with real data.

3

From Gueymard, the radiance of a partly cloudy sky can be considered as a weighted sum of the radiances of a clear and an overcast sky.

$$I_{D,\beta} = I_D\left[(1-N_{pt})r_{d0} + N_{pt}r_{d1}\right]$$

3

Here N_{pt} is the weighting factor and the subscripts 0 and 1 refer to the opacity (0 clear and 1 overcast). r_{do} is obtained as the sum of a circumsolar component (dependent on the angle of incidence θ) and a hemispheric component (dependent on the tilt of the collector β and the solar altitude α).

$$r_{d0} = \exp\left(a_0 + a_1 \cos\theta + a_2 \cos^2\theta + a_3 \cos^3\theta\right) + f(\beta)\, g(\alpha)$$

In this the parameters (themselves functions) were estimated and took the following forms:

$$a_0 = -0.897 - 3.364h + 3.960h^2 - 1.909h^3$$

$$a_1 = 4.448 - 12.962h + 34.601h^2 - 48.784h^3 + 27.511h^4$$

$$a_2 = -2.770 + 9.164h - 18.876h^2 + 23.776h^3 - 13.014h^4$$

$$a_3 = 0.312 - 0.217h - 0.805h^2 + 0.318h^3$$

$$f(\beta) = \left[1 + b_0 \sin^2(\beta) + b_1 \sin(2\beta) + b_2 \sin(4\beta)\right] / \left[1 + b_0\right]$$

$$g(\alpha) = 0.408 - 0.323h + 0.384h^2 - 0.170h^3$$

In the last few relationships

$$h = 0.01\alpha \quad b_0 = -0.2249 \quad b_1 = 0.1231 \quad b_2 = -0.0342.$$

4- Perez model:

Provides predictions of the PV energy and the module temperature with a difference up to 3%.

The Perez et al. model is based on a three components treatment of the sky diffuse irradiance. The incident diffuse energy on any tilted surface is given by:

$$I_{D,\beta} = I_D \left[(1 - F_1) \cos^2 \left(\frac{\beta}{2} \right) + F_1 \left(\frac{a_0}{a_1} \right) + F_2 \sin(\beta) \right]$$

Here, F_1 and F_2 are circumsolar and horizon brightening coefficients, and a_0, a_1 account for the respective angles of incidence of circumsolar radiation on the tilted and horizontal surfaces.

$$a_0 = \max[0, \cos(\theta)], \ a_1 = \max[\cos(85^{\circ}), \cos(Z)]$$

With these definitions, $a_0/a_1 = r_B$. The brightness coefficients F_1, F_2 are functions of the zenith angle Z, sky clearness ε_i, and a brightness Δ. ε_i is a function of I_D and the normal beam irradiation $I_{B,n}$.

$$\varepsilon_i = \left\{ \left[I_D + \frac{I_{B,n}}{I_D} \right] + 5.535 \times 10^{-6} Z^3 \right\} / \left\{ 1 + 5.535 \times 10^{-3} \right\}$$

Note that in this equation, Z is in degrees.

$\Delta = mI_D/I_{E,n}$

where m is air mass and $I_{E,n}$ is the extraterrestrial normal incidence irradiation.

$F_1 = \max\left[0, F_{11} + F_{12}\Delta + \left(\frac{\pi}{180}\right) ZF_{13}\right]$

$F_2 = \left[F_{21} + F_{22}\Delta + \left(\frac{\pi}{180}\right) ZF_{23}\right]$

The F_{ij} are obtained from Table 1 for various values of ε_i.

Table 1. Coefficients used in the Perez et al. [6] model.

ε_i (bins)	1	2	3	4	5	6	7	8
Lower Bound	1.000	1.065	1.230	1.500	1.950	2.800	4.500	6.200
Upper Bound	1.065	1.230	1.500	1.950	2.800	4.500	6.200	–
F_{11}	−0.0083	0.1299	0.3297	0.5682	0.8730	1.1326	1.0602	0.6777
F_{12}	0.5877	0.6826	0.4869	0.1875	−0.3920	−1.12367	−1.5999	−0.3273
F_{13}	−0.0621	−0.1514	−0.2211	−0.2951	−0.3616	−0.4118	−0.3589	−0.2504
F_{21}	−0.0596	−0.0189	0.0554	0.1089	0.2256	0.2878	0.2642	0.1561
F_{22}	0.0721	0.0660	−0.0640	−0.1519	−0.4620	−0.8230	−1.1272	−1.3765
F_{23}	−0.0220	−0.0289	−0.0261	−0.0140	0.0012	0.0559	0.1311	0.2506

• ***Hay-Davies and Perez show lower deviation than other models; similar to each other.***

Photovoltaic (PV) system performance evaluation is a complicated task or process. To assess the PV system generation, several climatic data sources are available. Different computational models may be used to calculate solar irradiance on the array's plane (POA) and many meteorological datasets can be used as data input for prediction models.

The PV module's cooling impact due to wind speed should not be overlooked.

-Study for Perez and Hay-Davies models:

Perez and Hay-Davies models for the computing of the irradiance on tilted surfaces are combined with three meteorological datasets, characterized by different monitoring period and meteo station location, to estimate the POA irradiance, the module temperature and PV energy output for a PV system located in the Mediterranean climate area. The PVsyst tool does the prediction and compares the findings to the actual data. Wind influences on PV module performance are taken into consideration during simulations.

In general, meteorological data sources collect hourly or monthly values of global horizontal irradiance, direct horizontal irradiance, ambient temperature, and wind speed for a specified place in a specific time. PVsyst also allows you to import hourly meteo data for a whole year in an appropriate format. There are three meteorological datasets used for the simulations: Meteonorm (MTN), data provided by the World Meteorological Organization (WMO) and data by the Regional Agency for environmental Prevention and Protection (ARPA).

Meteorological database	Monitoring period	Geographic area of meteo station location	Distance from meteo station to PV system
MTN	1991–2010	Rural area	>40 km
WMO	1951–1970	Rural area	<10 km
ARPA	2010–2013	Urban area	<10 km

* Table: Meteo datasets features.

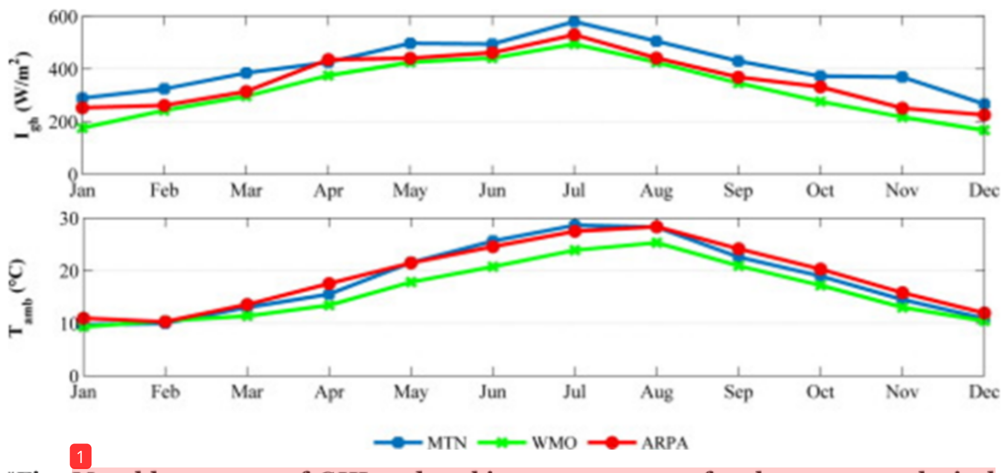


*Fig: Monthly average of GHI and ambient temperature for three meteorological datasets.

-PVsyst simulations:

A brief description of the simulation process implemented in PVsyst is presented in the following. It involves three main steps:

1.Determination of the effective incident global irradiation on the plane of the array;

2.Determination of the effective energy at the array output;

3.Determination of the energy output.

When the simulations are running in PVsyst tool, global horizontal irradiance, ambient temperature and wind speed hourly data are read from the selected meteo dataset to perform the results for each hour of the year.

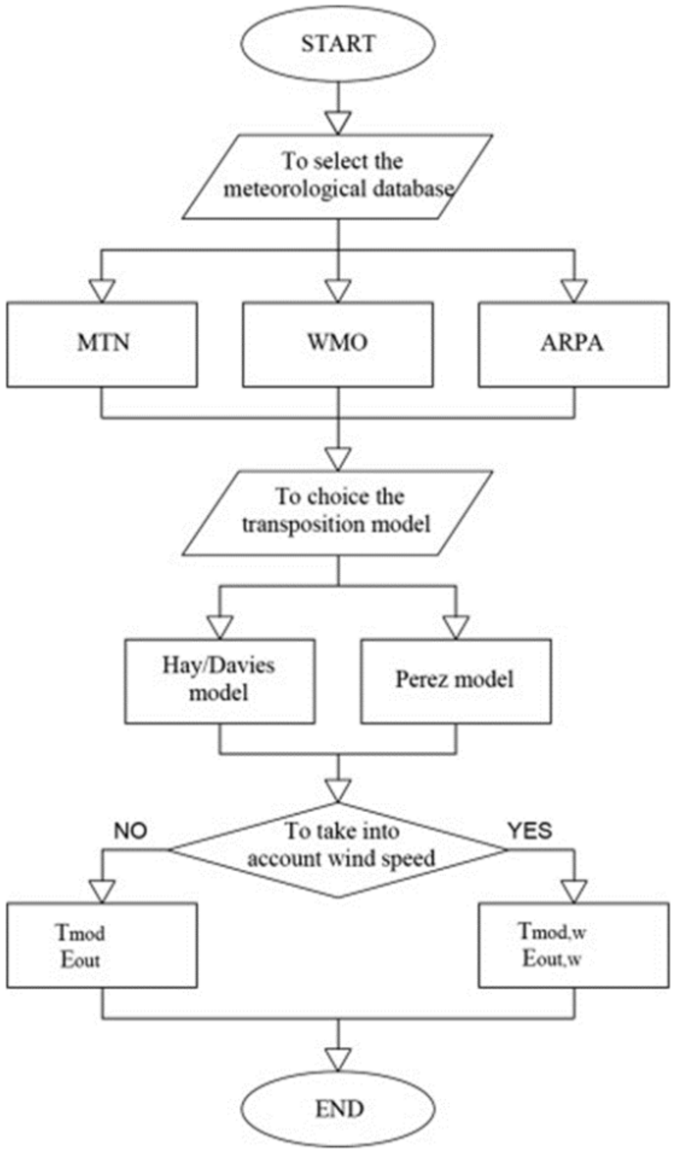


*Fig: Flow chart of PVsyst simulation process.

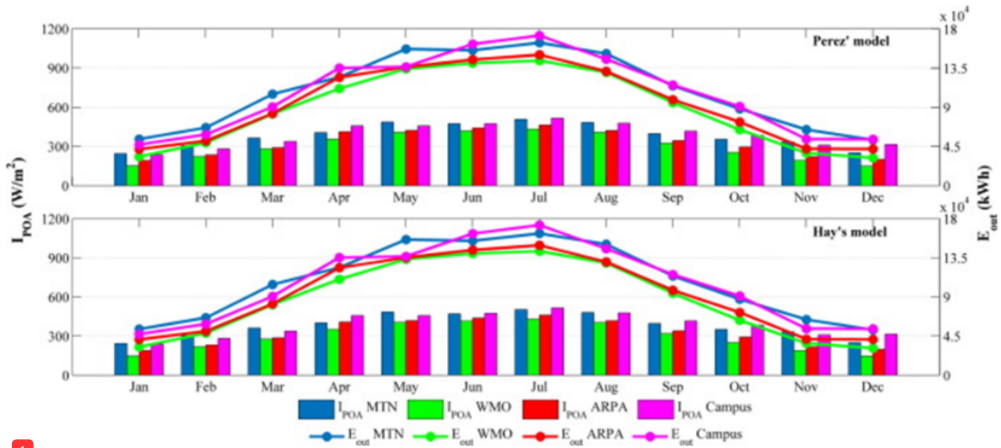
The POA irradiance was calculated using the PVsyst program for three distinct meteorological databases connected to the site examined, using both Hay-Davies and Perez transposition models. The module temperature (Tmod) and PV energy output (Eout) were calculated with and without the influence of wind speed in each example. The flow chart depicted in the previous graphic reflects all of PVsyst's situations.

- Discussion of simulation findings

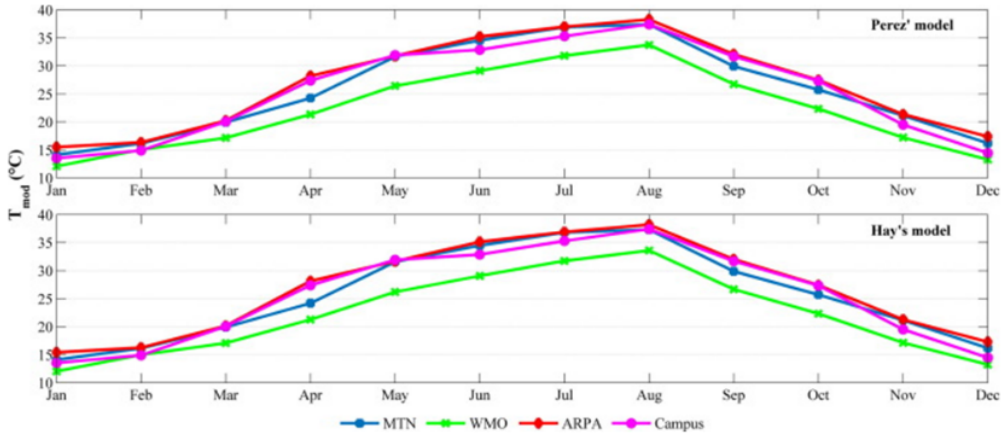
This section summarizes the key findings of a comparison of PVsyst's POA irradiance, module temperature, and PV energy production projections with actual observed data for the year 2013.

- Simulation outcomes and error analysis

The two following figures shows estimations of POA irradiance, PV energy, and module temperature for three meteorological datasets using both Hay-Davies and Perez transposition models against measured data in campus.



*Monthly average of estimated vs measured POA irradiance and PV energy for Perez and Hay-Davies models.



***Monthly average of estimated vs measured module temperature for Perez and Hay-Davies models.**

Generally, the Perez model gives higher value than the Hay-Davies model in the range 0.75–2.8% for all parameters. The Hay-Davies and Perez transposition models, which are used to determine the POA irradiance, have a greater influence on PV energy predictions than on module temperature estimations. The difference between the two models in PV energy estimates is up to 3%, and in temperature predictions it is up to 1%, with a large discrepancy during the winter. Our findings show that the PV energy projected by the Perez model is up to 2.75 percent higher in cold months and up to 1% higher in hot months than the Hay-Davies model. Nonetheless, the Perez transposition model produces PV energy production estimates that are more accurate than the Hay-Davies model.

- **Hay-Davies model:**

In the Hay-Davies model the total irradiance on plane of array I_{POA}.

is given by:

$$I_{POA} = (I_b + I_d A_i) R_b + \left[I_d (1 - A_i) \left(\frac{1+\cos\beta}{2} \right) \cdot I \rho_g \left(\frac{1-\cos\beta}{2} \right) \right]$$

where A_i is an anisotropy index, R_b is the geometric factor, β is the tilt angle and ρ_g is the ground reflectance.

The Perez model estimates the total irradiance on plane of array I_{POA} as follows:

$$I_{POA} = I_b R_b + I_d (1 - F_1) \left(\frac{1+\cos\beta}{2} \right) + I_d F_1 \frac{a}{b} + I_d F_2 \sin\beta I \rho_g \left(\frac{1-\cos\beta}{2} \right)$$

Where F_1 is the circumsolar brightness coefficient and F_2 is the horizon brightness coefficient, $a = \max\{0, \cos\theta\}$ and $b = \max\{\cos 85, \cos\theta_z\}$, with θ the angle of incidence and θ_z the zenith angle.

Since the sun position differs throughout the year, optimum PV panel tilt angle must be differing as well. That is why -monthly varying- angle value is also integrated into the fundamental part of the equation that are related with the solar diffusion and the reflectance effects.

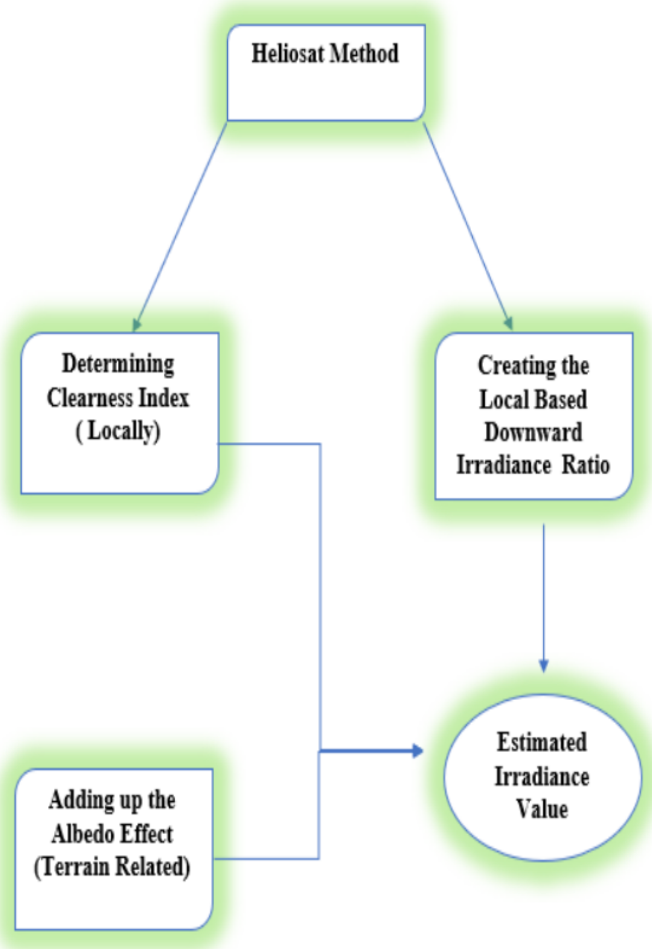
Heliosat Method:

Heliosat method basically utilizes satellite weather information in order to rebuild them as a location based data that will be used in solar energy calculations.

Since the meteorological data that will be taken into consideration is gathered from the NASA originated resource, it appears to be that one of the most appropriate and applicable method is Heliosat Method.

Some of the input data may follow as;

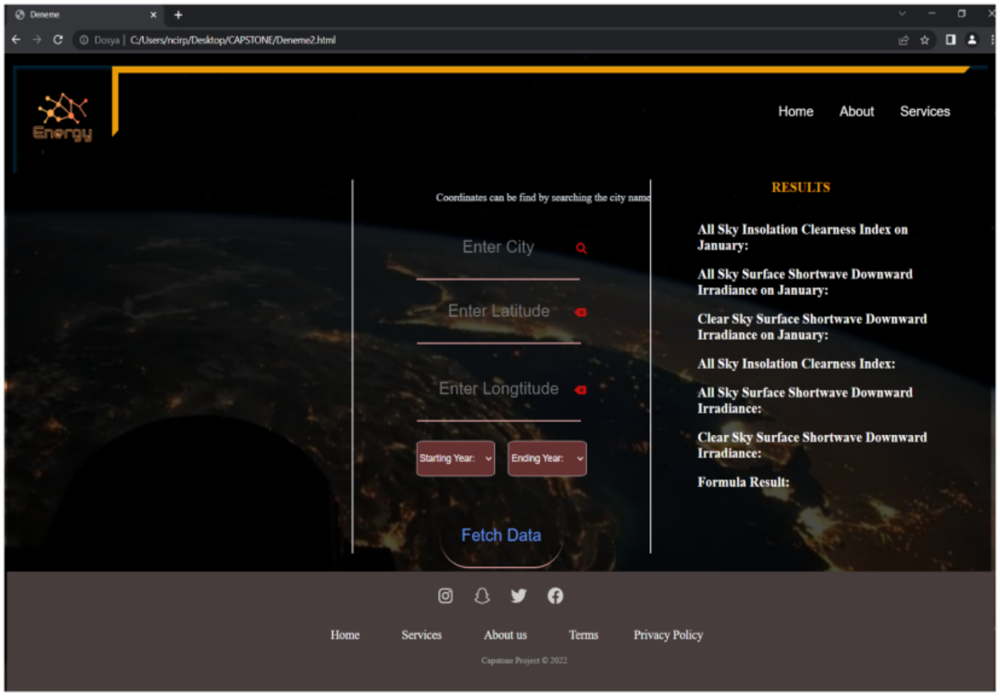
- **Clearness Index**
- **Clear Sky Surface Downward Irradiance**
- **Clear Sky Surface Downward Direct Normal Irradiance**
- **All Sky Surface Downward Irradiance**
- **All Sky Surface Downward Direct Normal Irradiance**
- **Ground Reflectance Irradiance (Albedo)**



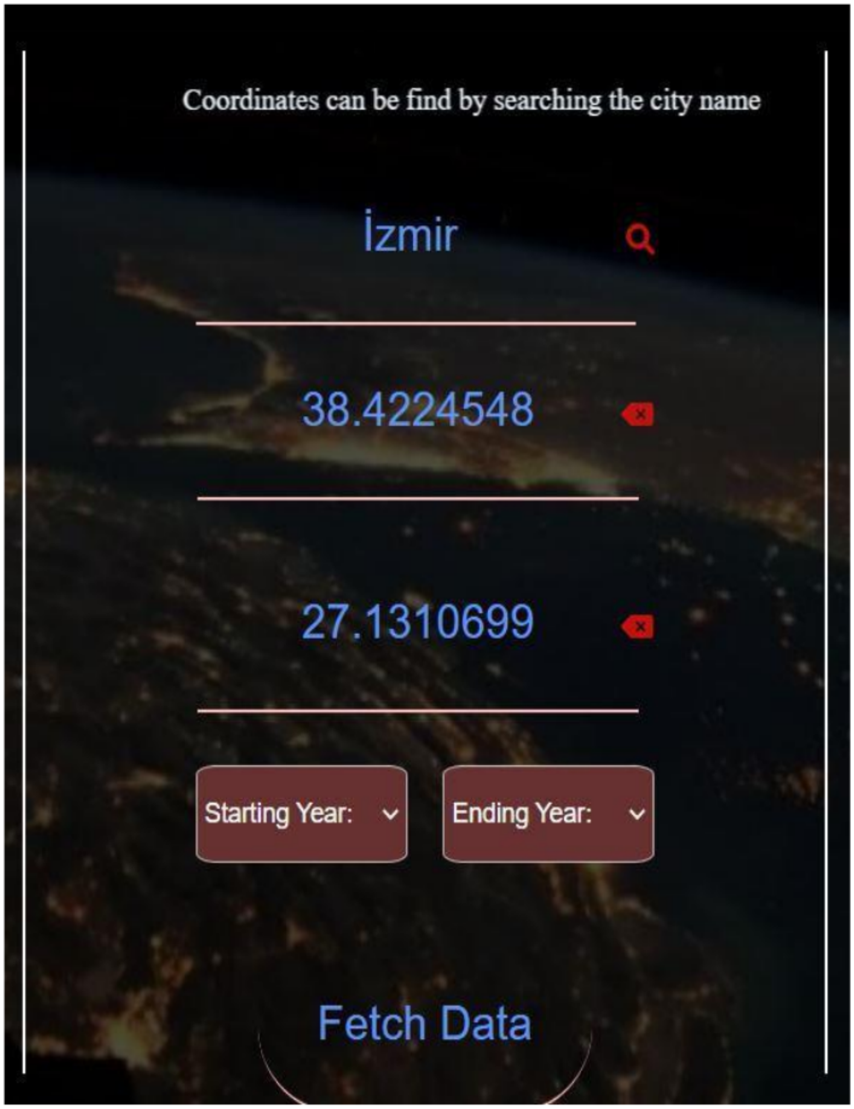
Roadmap through creating the formula structure

Interface Design

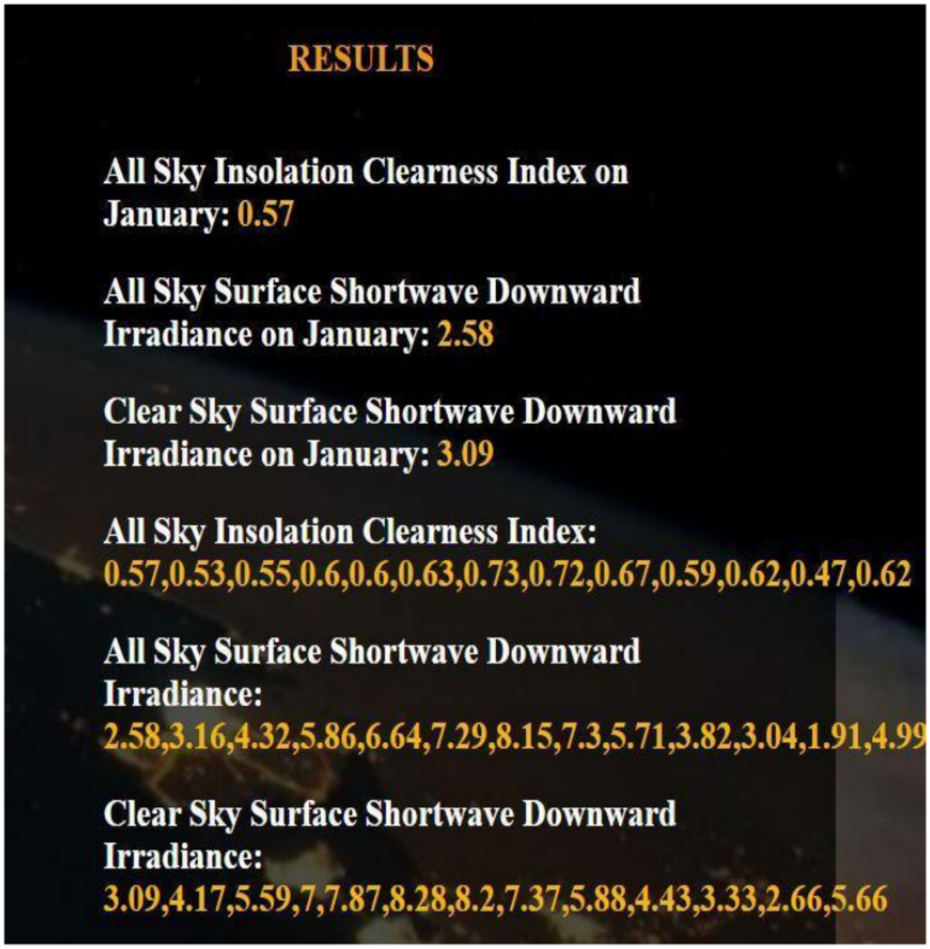
Old Version :



Home Page: It is the page where the main functions will be performed. Users can view some information about the site from the about section on this page. As the main function, the user enters the coordinates of the place on this page where the user wants to view his data. If the user does not know the coordinates, the user enters the name of the city he/she wants to search in the "city" section and the coordinates are displayed by the system. With the "Fetch" button, the data at that coordinates entered by the user are retrieved from NASA's website and presented to the user on a monthly basis. The retrieved data consists of 3 separate parameters. These 3 parameters are "All Sky Insolation Clearness Index", "All Sky Surface Shortwave Downward Irradiance" and "Clear Sky Surface Shortwave Downward Irradiance" parameters.

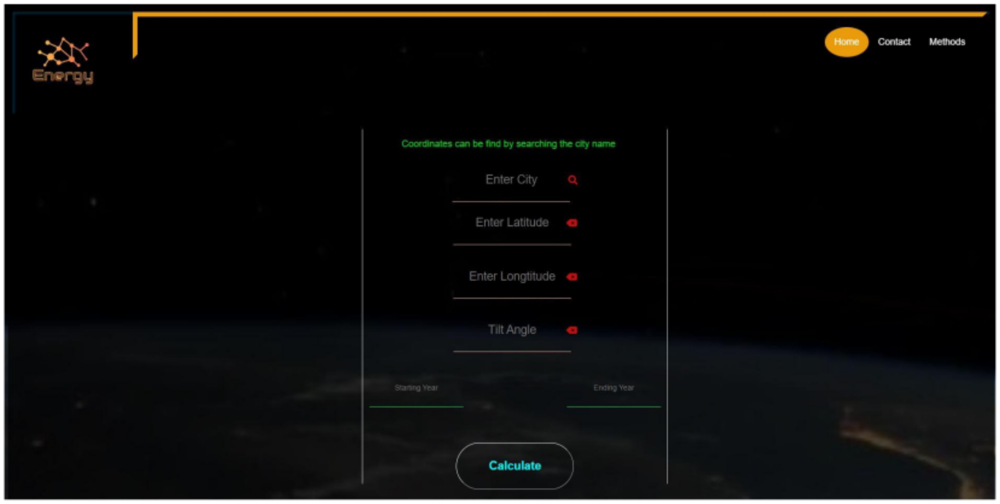


As an example, the city of Izmir and that coordinates of the city are shown. The user can view the coordinates of the city by typing the name of the city and using the search button on the side.

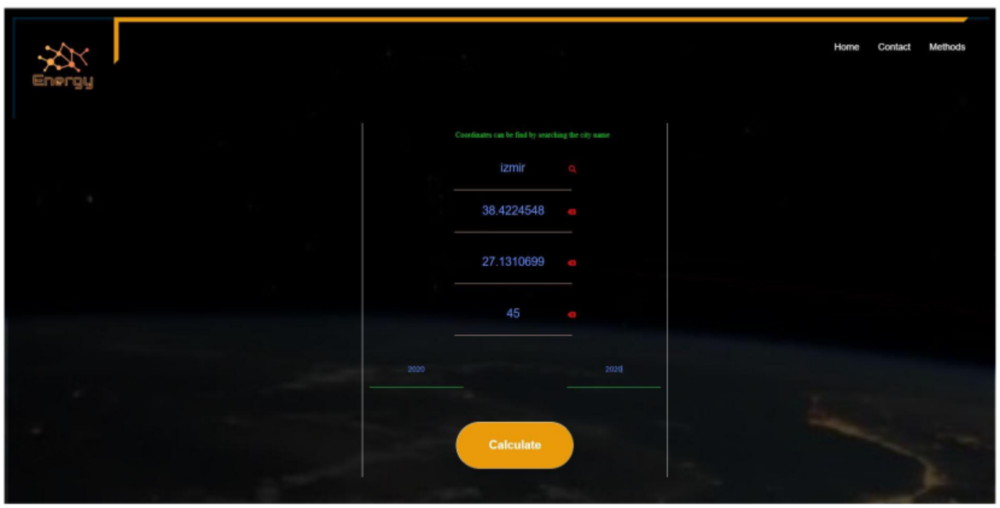


Results: The data retrieved from NASA is presented to the user on a monthly basis with its parameters. Monthly data is displayed in sequential form.

Final Version :

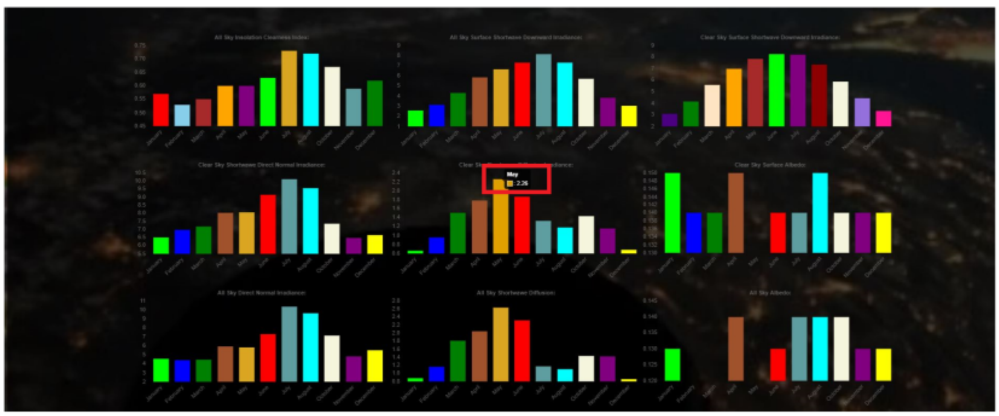


Added tilt angel function. The user must enter the angle value. The formula calculates accordingly. The year function has started working.



The necessary data for a sample calculation has been entered into the system. The city of Izmir was entered as an example location. Then the coordinates of that city are added automatically. The user has entered a 45 degree angle. The year 2020 was chosen.

According to the user's inputs, instant data is retrieved and calculated by the system. In this process, the "calculate" button has turned into a loading icon.It is indicated within the red rectangle in the picture.

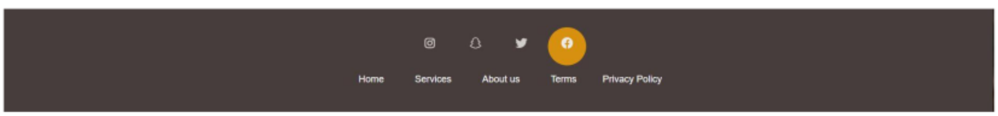


According to the inputs, 9 different parameters were taken instantly from nasa.com.These parameters are:

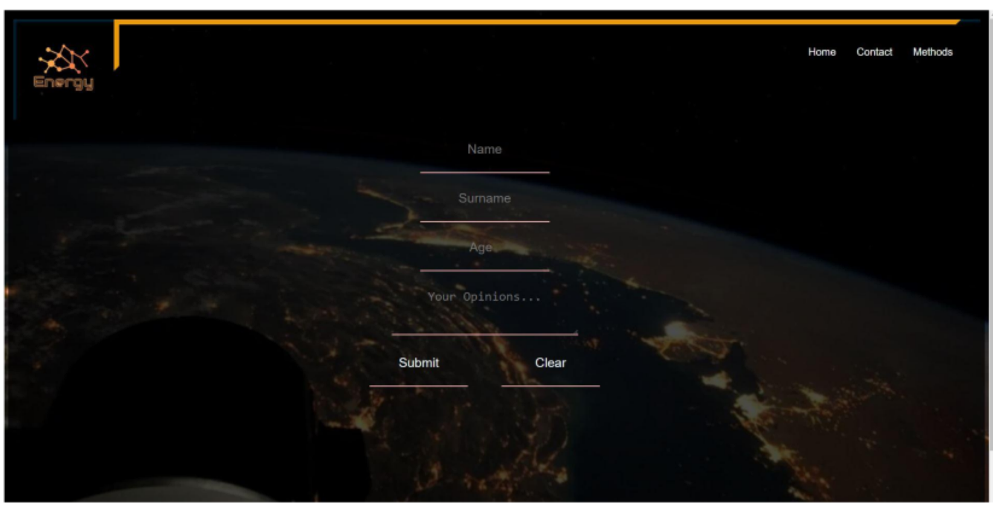
1. All Sky Insolation Clearness Index
2. Clear Sky Surface Shortwave Downward Irradiance
3. All Sky Surface Shortwave Downward Irradiance
4. Clear sky Shortwawe Direct Normal Irradiance
5. Clear Sky Shortwawe Diffusion Irradiance
6. Clear Sky Surface Albedo
7. All Sky Direct Normal Irradiance
8. All Sky Shortwawe Diffusion
9. All Sky Albedo

The user can view the full value by moving mouse over the bar of the month user wants to see. The picture shows the exact value of the "Clear Sky Shortwave Diffusion Irradiance" parameter for May.

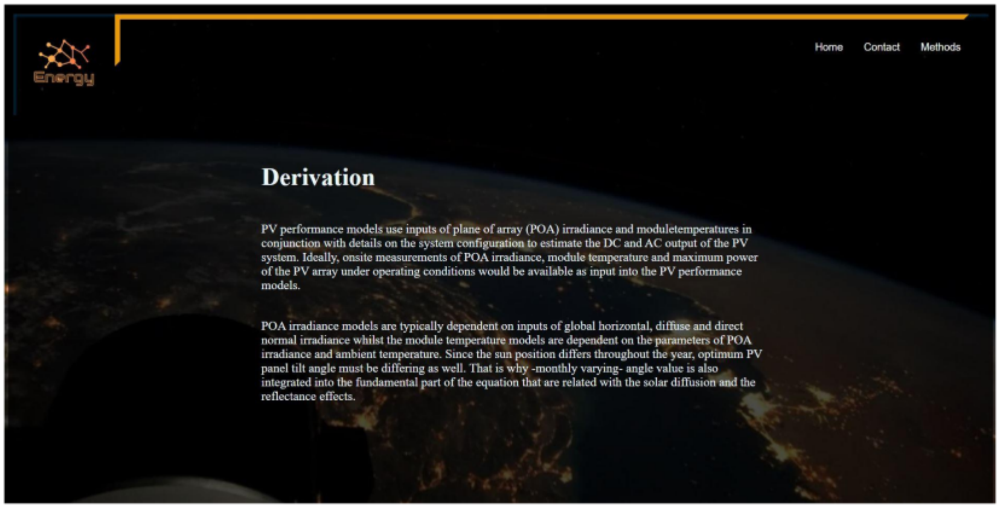
As a result of the calculation, the data is shown to the user monthly. The user can view the full value by moving mouse over the month user wants to see.



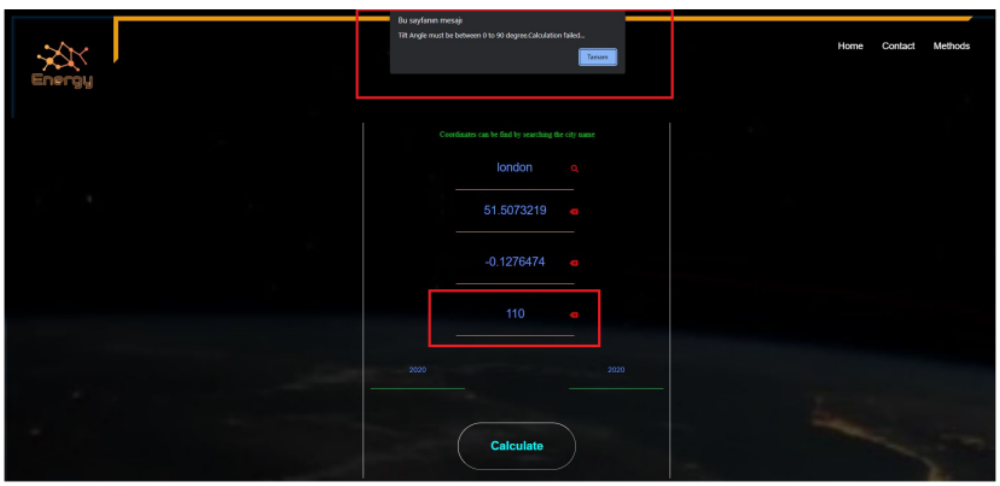
This part is the footer part of the page. The user accesses the relevant page by clicking on the option user wants to view.



The user can express their opinions about the site on this page. For this, the user's name, surname, age and opinion are requested.



On this page, some information is given to the user. This information relates to the methods the system uses for calculation. The user can view this page with the "methods" option from the menu.

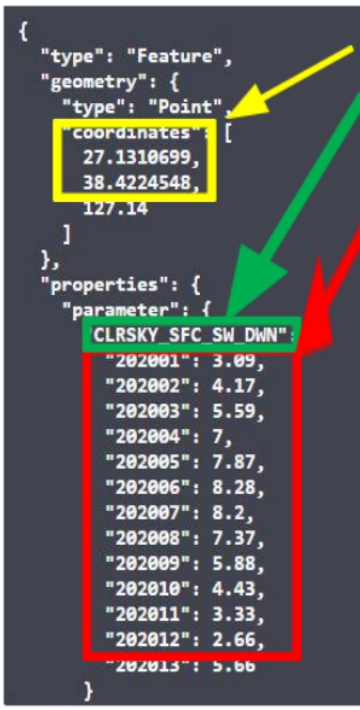


On this homepage, the user has to set the tilt angel value between 0-90. Otherwise, the site will show an error message to the user and no calculation will be made. As seen in the picture, the site gave an error message because the user entered the "tilt angel" value of 110.

Code Design

Old Version :

```
async function getData_CLRSKY_SFC_SW_DWN()
{
    var latitude = document.getElementById("latitude").value;
    var longitude = document.getElementById("longtitude").value;
    const api_url_2nd_parameter ='https://power.larc.nasa.gov/api/temporal/monthly/point?start=2020&end=2020&latitude='+latitude+'&longitude='+longitude+'&community=re
    const response = await fetch(api_url_2nd_parameter);
    const data = await response.json();
    const{properties}= data;
    const{parameter}= data.properties;
    const{CLRSKY_SFC_SW_DWN}= properties.parameter;
    array_CLRSKY_SFC_SW_DWN=Object.values(CLRSKY_SFC_SW_DWN);
    document.getElementById('CLRSKY_SFC_SW_DWN_January').textContent =array_CLRSKY_SFC_SW_DWN[0];
    document.getElementById('CLRSKY_SFC_SW_DWN').textContent = array_CLRSKY_SFC_SW_DWN;
}
```



Coordinates

Parameter

The data for 12 months

The getData_CLRSKY_SFC_SW_DWN function allows retrieving data from the API with the parameter Clear Sky Surface Shortwave Downward Irradiance. Other functions have been created within other parameters. By entering the coordinate information by the user, a request is sent to the API for the Clear Sky Surface Shortwave Downward Irradiance parameter. As a result of the request, the API sends the data in JSON format. The 12-month Clear Sky Surface Shortwave Downward Irradiance parameter is retrieved from the JSON formatted data and displayed to the user. In the picture with JSON data, the entered coordinates and parameter results are shown.

```
async function getCoordinates()
{
    var city = document.getElementById("city").value;
    const api_url = 'https://api.opencagedata.com/geocode/v1/json?q='+city+'&key=54c402a5300d4371b1b15f0128864a4f&language=tr&pretty=1';
    const response = await fetch(api_url);
    const data2 = await response.json();
    const {results} = data2;
    const object_values=Object.values(results);
    var coordinates= object_values[0];
    const {geometry}= coordinates;
    const {lat,lng}= coordinates.geometry;
    var latitude_output = lat;
    var longtitude_output = lng;
    document.getElementById('latitude_output').textContent = latitude_output;
    document.getElementById('longtitude_output').textContent = longtitude_output;
    //alert("Latitude: "+latitude_output+"\nLongtitude: "+longtitude_output);
    document.getElementById("longtitude_output").style.display = 'none';
    document.getElementById("latitude_output").style.display = 'none';
    document.getElementById("latitude").value = latitude_output;
    document.getElementById("longtitude").value = longtitude_output;
}
```

Getting Coordinates: The getCoordinates function is the function that displays the coordinates of the city entered by the user. A request is sent to the API with the city entered. As a result of the request, the coordinates are retrieved from the opencagedata.com

Final Version

```
<head>
<link rel="stylesheet" href="Homepage.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.1/css/all.min.css">
<script src="https://kit.fontawesome.com/3a22411ee2.js" crossorigin="anonymous"></script>
<meta charset='utf-8'>
<meta http-equiv='X-UA-Compatible' content='IE=edge'>
<title>Homepage</title>
<meta name='viewport' content='width=device-width, initial-scale=1'>
<script src="Homepage.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"></script>
<script src ="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.1.1/css/fontawesome.min.css">

<script>
```

Added some new scripts.

```
<ul class="slider-menu">
    <a href="Homepage.html"> Home</a>
    <a href="Contact.html">  <li>Contact</li></a>
    <a href="Methods.html"> <li>Methods</li> </a>
</ul>
```

Necessary directions were given to the options in the menu.

```
<button onclick="getData_ALLSKY_KT(); getData_CLRSKY_SFC_SW_DWN(); getData_ALLSKY_SFC_SW_DWN();
getData_CLRSKY_SFC_SW_DNI(); getData_CLRSKY_SFC_SW_DIFF();getData_CLRSKY_SRF_ALB(); getData_ALLSKY_SFC_SW_DNI();
getData_ALLSKY_SFC_SW_DNI();getData_ALLSKY_SFC_SW_DIFF(); getData_ALLSKY_SRF_ALB();setTimeout(calculate, 9000)" class="button_fetch_data">Calculate</button>
```

New functions were created with new parameters and called in the "Calculate" button.

```
<div class="graph"> <!-- canvas -->
    <div class="c">
    <canvas id="All Sky Insolation Clearness Index:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
    <canvas id="All Sky Surface Shortwave Downward Irradiance:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
    <canvas id="Clear Sky Surface Shortwave Downward Irradiance:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="Clear Sky Shortwawe Direct Normal Irradiance:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="Clear Sky Shortwawe Diffusion Irradiance:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="Clear Sky Surface Albedo:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="All Sky Direct Normal Irradiance:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="All Sky Shortwawe Diffusion:"style="height:200; width:200;"></canvas>
    </div>
    <div class="c">
        <canvas id="All Sky Albedo:"style="height:200; width:200;"></canvas>
    </div>
    </div> <!-- canvas -->
    <div class="result_graph">
    <canvas id="results" style=" height: 250; width: 600;"></canvas>
</div>
```

Necessary divs were created to show the data retrieved from NASA to the user in graphic form.

```
<div class="footer"> <!-- Footer -->
        <footer>
        <div class="social"><a href="#"><i class="fa-brands fa-instagram"></i></a><a href="#"><i class="fa-brands fa-snapchat"></i></a><a href="#"><i class="fa-brands fa-twitter"></i></a
          <ul>
          <li><a href="#">Home</a></li>
          <li><a href="#">Services</a></li>
          <li><a href="#">About us</a></li>
          <li><a href="#">Terms</a></li>
          <li"><a href="#">Privacy Policy</a></li>
          </ul>
        </footer>
    </div> <!-- Footer End -->
```

Footer section added.

```
async function getData_ALLSKY_SFC_SW_DNI() // All Sky Direct Normal Irradiance
{
    var latitude =  document.getElementById("latitude").value;
    var longitude = document.getElementById("longtitude").value;
    var starting_date = document.getElementById("starting_date").value;
    var ending_date = document.getElementById("ending_date").value;
    const api_url_3rd_parameter ='https://power.larc.nasa.gov/api/temporal/monthly/point?start='+starting_date+'&end='+ending_date+'&latitude='+latitude+'&longitude='+longitude+'&community=re&par
    const response = await fetch(api_url_3rd_parameter);
    const data = await response.json();
    const{properties}= data;
    const{parameter}= data.properties;
    const{ALLSKY_SFC_SW_DNI}= properties.parameter;
    array_ALLSKY_SFC_SW_DNI=Object.values(ALLSKY_SFC_SW_DNI);

    for (var i = 0; i < 12; i++)
    global_Arr_6.push(array_ALLSKY_SFC_SW_DNI[i]);
    var xValues = ["January", "February", "March", "April", "May","June","July","August","October","November","December",];
    var yValues = [array_ALLSKY_SFC_SW_DNI[0],array_ALLSKY_SFC_SW_DNI[1],array_ALLSKY_SFC_SW_DNI[2],array_ALLSKY_SFC_SW_DNI[3],array_ALLSKY_SFC_SW_DNI[4],array_ALLSKY_SFC_SW_DNI[5],array_ALLSKY_S
    var barColors = ["lime", "blue","green","sienna","goldenrod","red","cadetblue","aqua","beige","purple","yellow"];

        new Chart("All Sky Direct Normal Irradiance:", {
          type: "bar",
          data: {
            labels: xValues,
            datasets: [{
              backgroundColor: barColors,
              data: yValues
            }]
          },
          options: {
            legend: {display: false},
            title: {
              display: true,
              text: "All Sky Direct Normal Irradiance:"
            }
          }
        });
}
```

Functions were created for the new parameters. As seen in the picture, a function has been created for the "All Sky Direct Normal Irradiance" parameter. In the new version of the functions, the start and end years are requested from the user. The data retrieved according to the inputs are in json format. The required data in json format is added to the globally defined array. Called and used in the "calculate" function. The necessary code structure has been created to show the data to the user in graphic form. The data in the array is added to the content of the chart with the for loop. Thus, the data from Nasa is shown to the user in the form of graphs on a monthly basis.

```javascript
function calculate()
{
  var latitude = document.getElementById("latitude").value;
   var z= parseInt(latitude);

   var t_angle = document.getElementById("t_angle").value;
   var t = parseInt(t_angle);
   if (t<0 || t>90 ) {
    alert("Tilt Angle must be between 0 to 90 degree.Calculation failed...");
}
    for(var i=0;i<12;i++){

        var m = i+1;
        var h= 0;
        if (m < 7) {
          h = [((m-1) * (7.67)) - 16.87]
        }
        else
        {
          h = 16.87 - ((m-7) * 7.67)
        }
        var k = z - h;
        var x = k - t;
        var g_direct_clear= global_Arr_3[i];
        var g_diffuse_clear = global_Arr_6[i] * (1+Math.cos(x)/2); // CLRSKY_SFC_SW_DIFF * (1+cos(x))/2
        var g_reflected_clear = global_Arr_1[i] * global_Arr_5[i] * (1-Math.cos(x)/2);
        var g_direct_all_sky = global_Arr_6[i];
        var g_diffise_all_sky = global_Arr_7[i] * (1+Math.cos(x)/2);
        var g_reflected_all_sky = global_Arr_2[i] * global_Arr_8[i] * (1-Math.cos(x)/2);
        var a = g_direct_clear+ g_diffuse_clear + g_reflected_clear;
        var b = g_direct_all_sky + g_diffise_all_sky + g_reflected_all_sky;
        var y = global_Arr_0[i];
        var final_result= (a*y)+ ((1-y)*b) ;
```

All calculations are done within this function. The latitude information in the city selected by the user or in the coordinates entered has been converted to integer data type. Tilt Angel data is retrieved from the user. The angle taken should be between 0-90. This condition is provided by the if structure. The "math" library has been added so that trigonometric functions can be used.

```javascript
array_result.push(final_result);
var xValues = ["January", "February", "March", "April", "May","June","July","August","October","November","December",];
var yValues = [array_result[0],array_result[1],array_result[2],array_result[3],array_result[4],array_result[5],array_result[6],
var barColors = ["red", "green","blue","orange","brown","lime","purple","aqua","beige","cadetblue","brown","yellow"];

new Chart("results", {
  type: "bar",
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor: barColors,
      data: yValues
    }]
  },
  options: {
    legend: {display: false},
    title: {
      display: true,
      text: "RESULTS"
    }
  }
});
```

The calculation result is assigned to the variable "final_result". It is assigned to 12 separate data with the for loop. These data are added to the graph on a monthly basis.

```javascript
async function getData_CLRSKY_SRF_ALB() // Clear Sky Surface Albedo
{
    var latitude =  document.getElementById("latitude").value;
    var longitude = document.getElementById("longtitude").value;
    var starting_date = document.getElementById("starting_date").value;
    var ending_date = document.getElementById("ending_date").value;
    const api_url_3rd_parameter ='https://power.larc.nasa.gov/api/temporal/monthly/point?start='+starting_date+'&end='+ending_date+'&latitude='+latitude+'&longitude='+longitude+'&community=re&pa
    const response = await fetch(api_url_3rd_parameter);
    const data = await response.json();
    const{properties}= data;
    const{parameter}= data.properties;
    const{CLRSKY_SRF_ALB}= properties.parameter;
    array_CLRSKY_SRF_ALB=Object.values(CLRSKY_SRF_ALB);

    for (var i = 0; i < 12; i++)
    global_Arr_5.push(array_CLRSKY_SRF_ALB[i]);
    var xValues = ["January", "February", "March", "April", "May","June","July","August","October","November","December",];
    var yValues = [array_CLRSKY_SRF_ALB[0],array_CLRSKY_SRF_ALB[1],array_CLRSKY_SRF_ALB[2],array_CLRSKY_SRF_ALB[3],array_CLRSKY_SRF_ALB[4],array_CLRSKY_SRF_ALB[5],array_CLRSKY_SRF_ALB[6],array_C
    var barColors = ["lime", "blue","green","sienna","goldenrod","red","cadetblue","aqua","beige","purple","yellow"];

        new Chart("Clear Sky Surface Albedo:", {
          type: "bar",
          data: {
            labels: xValues,
            datasets: [{
              backgroundColor: barColors,
              data: yValues
            }]
          },
          options: {
            legend: {display: false},
            title: {
              display: true,
              text: "Clear Sky Surface Albedo:"
            }
          }
        });
}
```

The function of another newly added parameter.

```html
                <div class="logo"></div>
                <ul class="slider-menu">
                    <a href="Homepage.html"><li>Home</li></a>
                    <a href="Contact.html"><li>Contact</li></a>
                    <a href="Methods.html"> <li>Methods</li> </a>
                </ul>
            </div><!-- logo -->
            <form>
                <table border="2">
            <tr>

                <td colspan="1"><input type="text" name="name"  placeholder="Name" id="name" required></td>
            </tr>

            <tr>

                <td colspan="1"><input type="text" name="surname" id="surname" placeholder="Surname"required></td>
            </tr>

            <tr>

                <td> <input type="number" name="yas" id="age" min="18" max="100" placeholder="Age" ></td>
            </tr>



                <tr>

                    <td>
                        <textarea name="opion" placeholder="Your Opinions..."id="gorus"></textarea>
                    </td>
                </tr>
            </tr>

            <tr>
                <td colspan="2">
                <input style="color:■aliceblue; margin-right:30px;"type="submit" name="submit" id="kaydet" value="Submit">
                <input style="color:■aliceblue; margin-left: 30px;" type="reset" name="clear" id="sil" value="Clear">

                </td>
```

A form created on the "Contact" page to get the user's opinions about the site.

```html
Methods.html > html > body > div.transparent > div.header-bar > ul.slider-menu > a
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Methods</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='Methods.css'>
    <script> </script>
</head>
<body>
    <div class="transparent">
        <div class="header-bar">
                <div class="logo"></div>
                <ul class="slider-menu">
                    <a href="Homepage.html"><li>Home</li></a>
                    <a href="Contact.html"><li>Contact</li></a>
                    <a href="Methods.html"> <li>Methods</li> </a>
                </ul>
            </div><!-- logo -->
    <div class="text">
        <h1>Derivation</h1> <br>
         PV performance models use inputs of plane of array (POA) irradiance and moduletemperatures in conjunction with details on the system configuration to estimate the DC and AC output of the
        <p><br>POA irradiance models are typically dependent on inputs of global horizontal, diffuse and direct normal irradiance whilst the module temperature models are dependent on the paramet
        Since the sun position differs throughout the year, optimum PV panel tilt angle must be differing as well. That is why -monthly varying- angle value is also integrated into the fundamenta
    </div>
    </div>

</body>
</html>
```

The html structure of the "methods" page created to inform the user about the methods used in the calculations.

CSS Codes For Homepage

```css
body{
    background-image: url('Earth.png');
    background-repeat: no-repeat;
    background-size: cover;
}
.transparent{
  background: rgba(0, 0, 0,.7);
  height: 230vh;
  width: 100%;
}
.logo{
  background-image: url('logo.png');
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
  width: 20vh;
  height: 20vh;
  float: left;
}
.header-bar {
  width: 99%;
  height:20vh;
  border: 10px solid transparent;
  box-shadow: 4px 4px 5px rgba(40, 45, 48, 0.25) inset;
}
.header-bar::after {
  content: '';
  position: absolute;
  width: 85%;
  height: 80px;
  border: 10px solid transparent;
  transition: 0.3s linear all;
  border-top: 10px solid rgba(233, 155, 10);
  border-left: 10px solid rgba(233, 155, 10);
}
.slider-menu {
  position: absolute;
  top: 0;
  right: 0;
  display: flex;
  list-style-type: none;
```

```css
.slider-menu {
  position: absolute;
  top: 0;
  right: 0;
  display: flex;
  list-style-type: none;
  margin: 0;
  z-index: 1;
  margin-right: 50px;
  margin-top:50px ;
  padding-left: 50px;
}
.slider-menu > li {
  padding: 20px;
  font-family: 'Quicksand', sans-serif;
  color: #fff;
  cursor: pointer;
  user-select: none;
  font-size: 20px;
}
.slider-menu > li:hover {
  color: rgb(255, 255, 255);
  background-color: #e99b0a;
  border-radius: 50%;
}
a {
  padding: 20px;
  font-family: 'Quicksand', sans-serif;
  color: #fff;
  cursor: pointer;
  user-select: none;
  font-size: 20px;
  text-decoration: none;
}
a:hover{
  color: rgb(255, 255, 255);
  background-color: #e99b0a;
  border-radius: 50%;
}
```

```css
.middle{
  height: 70vh;
  width:200vh;
}
.middle_left{
  align-items:center;
  text-align:center;
  height:70vh;
  width:60vh;
  margin-left:35%;
  float:left;
  border-left: 2px solid rgb(255, 255, 255);
  border-right: 2px solid rgb(255, 255, 255);

}
.middle_left input{
    width:190px;
    padding:30px;
    background:transparent;
    border:0;
    outline:0;
    border-bottom: 2px solid #fbb;
    text-align: center;
    font-size: 25px;
    color: cornflowerblue;
}
.middle_right{
  float: right;
  width: 50vh;
  height: 70vh;
  margin-right: 5%;
}
.button_fetch_data{
  width:250px;
  height:100px;
  font-weight: bold;
  text-align: center;
  text-decoration: none;
  border-radius: 50px;
```

```css
}
.active{
  font-size: 0;
  width:70px;
  height:70px;
  border-radius: 50%;
  border-left-color: transparent;
  animation: rotate 1.5s ease 0.5s infinite;
}
@keyframes rotate{
  0% {
    transform: rotate(360deg);
  }
}
```

```css
  .year {
    padding:2%;
    height:50px;
  }
  #starting_date{
    float: left;
    width: 13vh;
    font-size:medium;
    border-bottom: 2px solid rgb(9, 237, 100);
  }
  #ending_date{
    float: right;
    width: 13vh;
    font-size:medium;
    border-bottom:2px solid rgb(9, 237, 100);
  }
  .fa-1x{
    color:rgb(197, 16, 16);
    margin-left: -10px;
  }
  /*-- Middle End--*/
.output_label{
color:rgb(255, 255, 255);
 font-size:20px;
 font-weight: bold;
position:static;
}
.graph{
  width:155vh;
  height:25vh;
```

```css
.c{
  float:left;
  width: 50vh;
  height:25vh;
}
.result_graph{
  width: 100vh;
  height: 50vh;
  margin-left: 50vh;
  margin-bottom:5vh;
}
 /*-- Footer--*/
 .footer {
  padding:40px;
  background-color:#473d3d;
  color:#ffffff;
  width:95%;
}
.footer ul {
  list-style:none;
  text-align:center;
  font-size:18px;
}
.footer li {
  padding:0 10px;
  display: inline;
}
```

```css
.footer ul a {
  color:inherit;
  text-decoration:none;
  padding: 20px;

}
.footer ul a:hover {
  background-color: #e99b0a;
  border-radius: 50%;

}
.footer .social {
  text-align:center;
}
.footer .social > a {
  font-size:24px;
  width:40px;
  height:40px;
  display:inline-block;
  margin-top: 20px;
  margin:0 8px;
  color:inherit;
  opacity:0.75;
}
.footer .social > a:hover {
  opacity:0.9;
}
.footer .copyright {
  margin-top:15px;
  text-align:center;
  font-size:13px;
  color: #aaa;
  margin-bottom:0;
}
 /*-- Footer End--*/
p{
  -moz-animation: marquee 10s linear infinite;
  -webkit-animation: marquee 10s linear infinite;
  animation: marquee 10s linear infinite;
```

```css
@-moz-keyframes marquee {
  0% {
    transform: translateX(0%);
  }
  100% {
    transform: translateX(50%);
  }
}
@-webkit-keyframes marquee {
  0% {
    transform: translateX(0%);
  }
  100% {
    transform: translateX(50%);
  }
}
@keyframes marquee {
  0% {
    -moz-transform: translateX(0%);
    -webkit-transform: translateX(0%);
    transform: translateX(15%)
  }
  100% {
    -moz-transform: translateX(0%);
    -webkit-transform: translateX(0%);
    transform: translateX(-10%);
  }
}
```

Development Steps:

1)

1.1 We created the basic design structure of the site.

1.2 We created the demo that performs the basic functions.

1.3 We found the APIs of the sites we wanted to extract data from and integrated them into our site.

1.4 We sent the request through the API and retrieved the data.

1.5 We made a demo for basic calculation operations..

2)

2.1 The final version of the formula will be integrated into the project.

2.2 The system will be tested for the accuracy of mathematical operations.

2.3 The site design will be finalized.

Resources

https://www.sciencedirect.com/science/article/pii/S0196890416308688?casa_token=ua9QL8fQBnAAAAAA:UZHwiUeTTqMaw-PF1cGDzSgNnUjnq0cgEYMIV7xwC9S_bxmthIbnAQy38XISPRb36eQTjXt6aQ#b0175

https://www.sciencedirect.com/science/article/pii/S0960148115302846?casa_token=LKGk90tEpCAAAAAA:MH6ZL7SKKRrzeALH-I7HEb7jJNpXOM4mzIkpEfLDWmoVHsjkkteHtpwTtl3-svaRwE1Fz2zgOKo#bib1

https://www.sciencedirect.com/science/article/pii/S0960148112005447

https://kmu.edu.tr/akademikenerji

Marius Palescu et al., Predictive Modelling for Energy Management and Power Systems Engineering, 2021

4992Report1010283.pdf

ORİJİNALLİK RAPORU

%16	%6	%11	%5
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	M. Malvoni, M.C. Fiore, G. Maggiotto, L. Mancarella, R. Quarta, V. Radice, P.M. Congedo, M.G. De Giorgi. "Improvements in the predictions for the photovoltaic system performance of the Mediterranean regions", Energy Conversion and Management, 2016 Yayın	%7
2	Submitted to Bahcesehir University Öğrenci Ödevi	%2
3	hal.archives-ouvertes.fr İnternet Kaynağı	%2
4	Copper, J.K., A.B. Sproul, and S. Jarnason. "Photovoltaic (PV) performance modelling in the absence of onsite measured plane of array irradiance (POA) and module temperature", Renewable Energy, 2016. Yayın	%1
5	Practical PHP and MySQL Web Site Databases, 2013. Yayın	%1
	Submitted to CSU, San Jose State University	

6	Öğrenci Ödevi	%1
7	Mathieu David, Philippe Lauret, John Boland. "Evaluating tilted plane models for solar radiation using comprehensive testing procedures, at a southern hemisphere location", Renewable Energy, 2013 Yayın	%1
8	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%1
9	Submitted to University of Nottingham Öğrenci Ödevi	%1

Alıntıları çıkart	Kapat	Eşleşmeleri çıkar	< %1
Bibliyografyayı Çıkart	üzerinde		