Using React and Fuzzy Expert Systems for Better Travel Experience in Local Route Planning

Nataliia Kunanets [1], Olga Artemenko [2], Volodymyr Pasichnyk [1] , Vasyl Kut [3] and Arsen Ivanchov [3]

[1] *Lviv Polytechnic National University, Stepana Bandery Street 32-a, Lviv, 79013, Ukraine*
[2] *PHEI Bukovinian University, Darvina Street 2-A, Chernivtsi, 58000, Ukraine*
[3] *Uzhhorod National University, Narodna Square 3, Uzhorod, 88000, Ukraine*

Abstract

In this research the design and construction of a cross-platform recommender application for the optimization of a local touristic route planning is described. The subject area is limited to the tourist routes of the Transcarpathia region in order to improve the user experience at the local level. An aspect of the approach proposed by the authors is the optimization of the tourist route by supplementing it with points of interest based on the assessment of their recreational attractiveness. Fuzzy modeling is used to evaluate the recreational attractiveness of a specific tourist object or object of social infrastructure. Public cartographic services were used as a source of geoinformation data. The web application created as part of the research allows the user to visualize on a digital map the places of interest for the tourist and several options for the route. The React library was used to implement the software product.

Keywords

Route planning, tourism, travel experience, React, digital mapping services.

1. Introduction

The purpose of developing this application was to study new technologies and demonstrate the use of modern information solutions to improve the user experience in the field of planning tourist routes. Today's world has entered a new stage of life, where information plays the main role. The modern development of the information society is directly related to the need to collect, process and transfer huge volumes of information, transforming information into a product, as a rule, of considerable value [1-3]. This became the reason for the global transition from an industrial society to an information society. The emergence of the worldwide Internet caused a large-scale growth of international communications in various spheres of human life.

In terms of the level of development of Internet technologies, Ukraine is among the leaders of dozens of countries in the world. Today, information technologies occupy an important place in the tourism sphere of Ukraine. Information technologies have given new opportunities for work and rest, in many ways have made the work and simple life of every modern person easier. Today's society can hardly be imagined without information technologies that allow planning a tourist trip, both in Ukraine and abroad. Planning should be an integral part of any trip in which the traveler wants to meet the expectations of the vacation and get aesthetic and cognitive pleasure [4, 5]. Travel planning in Ukraine is key to supporting sustainable tourism development, and while some destinations are already very well developed, others need better research.

MoMLeT+DS 2023: 5th International Workshop on Modern Machine Learning Technologies and Data Science, June 3, 2023, Lviv, Ukraine
EMAIL: nek.lviv@gmail.com (N. Kunanets); olga.hapon@gmail.com (O. Artemenko); vpasichnyk@gmail.com (V. Pasichnyk); kut.vasilij81@gmail.com (V. Kut); ivanchov.arsen@student.uzhnu.edu.ua (A. Ivanchov)
ORCID: 0000-0003-3007-2462 (N. Kunanets); 0000-0002-4057-1217 (O. Artemenko); 0000-0001-9434-563X (V. Pasichnyk); 0000-0001-5267-331X (V. Kut); 0009-0000-7898-7667 (A. Ivanchov);

At the basic level, the development of tourism in the Transcarpathian region can be defined as the creation of strategies and plans for the increase/development/encouragement of this industry. The main reason for planning and implementing strategies for the development of the tourism sector is, first of all, earning and further increasing the GDP of the country/region. By planning your trip, you can gain some flexibility in case of unforeseen circumstances during the trip (weather, road accidents, etc.) and avoid wasting time searching the Internet for places worth visiting. Thus, by putting a little effort into planning, you can optimize the useful time in your trip and get the maximum of impressions.

The purpose of the study is to study new technologies and demonstrate the use of modern information solutions to improve user experience in the field of tourism planning [6-8]. Based on the purpose of the research, the following tasks are set in the work:

- Conduct an analysis of tourist attractiveness in Zakarpattia Oblast.
- Analyze the possibilities of using services such as GoogleMaps for data analysis in own applications.
- To analyze the React JavaScript library, its capabilities and features, as a tool for the development of tourism web applications.
- To create a recommender system based on location data and fuzzy expert system to improve local route planning [9].
- Develop a web application for planning tourist routes in Transcarpathia. The object of research is the process of developing web applications using open mapping services. The subject of the research is the creation of a web application for planning tourist routes in Transcarpathia using the React library.

2. Using free access digital cartographic services and visualization technologies

GoogleMaps has more coverage than OpenStreetMap(OSM), making it the leading map in many countries, such as Japan or Ukraine. OpenStreetMap has less coverage, but the user can edit the points to add information. GoogleMaps has detailed coverage down to the smallest streets. Using GoogleMaps, you can even find local names of locations.

GoogleMaps provide an attractive color and design, thanks to which you can easily distinguish different elements on the map [10-12]. OpenStreetMap has several color schemes for highlighting features, so it takes time to distinguish features.

Apart from these differences, the similarity between these services is that they both have an API that allows users to access maps and data on web pages and in other applications. The GoogleMaps API uses a simple JavaScript interface to allow developers to easily embed GoogleMaps into their web application. OpenStreetMap also allows you to access the data using the Overpass API, which is a read-only API. The user can send requests and receive datasets.

OSM provides better performance in various urban centers compared to GoogleMaps. GoogleMaps service is best for online purposes. OpenStreetMap allows users to upload maps that can be used offline to find a route or find a location.

Both have many advantages and disadvantages over each other, but it completely depends on the user's preference as to which one suits them best. The choice depends on the specific needs of the user and availability React is a JavaScript library used to create user interfaces. It is a simple, feature-rich, component-oriented JavaScript library. React can be used both for the development of small programs and for large, complex web applications. This library provides a minimal and robust set of functions for running web applications.

API (Application Programming Interface) is a set of software code that provides data transfer between one software product and another. APIs allow one software product or service to interact with others without knowing how they are implemented. This can simplify the application development process and thereby save time and money. When developing new tools and applications or managing existing ones, the use of APIs gives the developer flexibility, simplifies the design, administration and use of software products.

3. Data processing and route planning algorithm

Before using the exported from GoogleMaps geodata to evaluate the recreational attractiveness of the territory, it must be analyzed and processed. For this, transformation algorithms were developed, the work of which is described in figure 1.

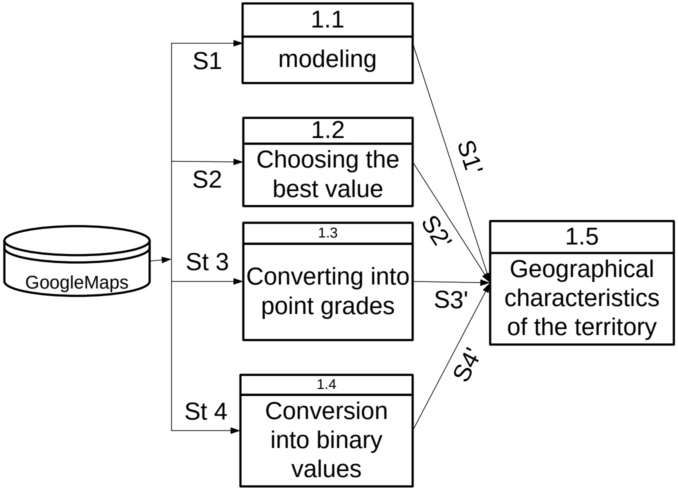

1.1
modeling
1.2
Choosing the best value
1.3
Converting into point grades
1.4
Conversion into binary values
1.5
Geographical characteristics of the territory
GoogleMaps
S1
S2
St 3
St 4
S1'
S2'
S3'
S4'

Figure 1: Scheme of data flows of the operation of loading geographic characteristics of the territory

Exported geodata layers are input to the algorithm in .dat file format. The contents of the file are converted into a vector of numerical values. The next step is the processing of geodata using modeling operations, conversion into point scores, selection of the best value and conversion into binary values To obtain information about places of interest (key points) for the route being laid, it was decided to use a third-party API called TravelAdvisor, its main function is to transfer information from the popular travel guide TripAdvisor [13] to the virtual address we need in the form of a JSON file.

For the obtained array of attractive points of the potential route and places of tourist infrastructure, it is necessary to evaluate their attractiveness rating for a specific user. For this purpose, it was decided to use the attractiveness indicator of the point of the tourist route, which is evaluated by fuzzy modeling as follows. Fuzzy modeling is often used for different touristic researches [14].

The indicator of the attractiveness of the point of the tourist route is presented as a linguistic variable P = "tourist attractiveness". The variable P is determined by the types of rest and recreation that can be organized and carried out in this area, which depend on the climatic, geographical, historical and cultural conditions and preferences of the user.

The attractiveness scale has five terms: "very low", "low", "medium", "high", and "very high". P ={"very low", "low", "medium", "high", and "very high"}={NB, NS, ZP, PS, PB}. We will represent each term as a membership function (Fig. 2)

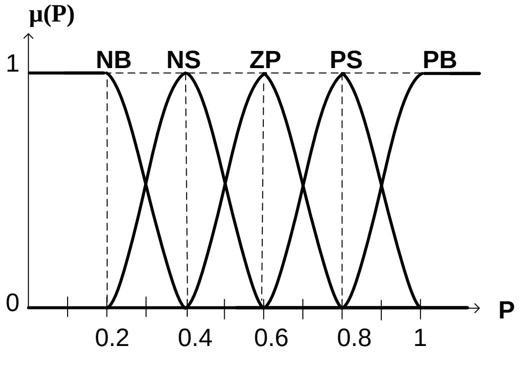


Figure 2: The membership functions of the terms of the linguistic variable P

The NB term is described by a Z-shaped membership function with parameters $(0;0.2)$:

$$\mu(P)=\begin{cases}1, & P\le 0\\ 1-2\left(\frac{P}{0.2}\right)^2, & 0<P\le 0.1\\ 2\left(\frac{0.2-P}{0.2}\right)^2, & 0.1<P<0.2\\ 0, & 0.2\le P\end{cases}. \tag{1}$$

The NS term is described by a membership function with parameters $(0.396;0.201)$:

$$\mu(P)=e^{\frac{-(P-0.201)^2}{2\cdot 0.396^2}} \tag{2}$$

The term ZP is described by a membership function with parameters $(0.5898;0.201)$:

$$\mu(P)=e^{\frac{-(P-0.201)^2}{2\cdot 0.5898^2}} \tag{3}$$

The PS term is described by a membership function with parameters $(0.8001;0.201)$:

$$\mu(P)=e^{\frac{-(P-0.201)^2}{2\cdot 0.8001^2}} \tag{4}$$

The term PB is described by an S-shaped membership function with parameters $(0.8;1)$of the following form:

$$\mu(P)=\begin{cases}0, & P\le 0.8\\ 2\left(\frac{P-0.8}{0.2}\right)^2, & 0.8<P\le 0.9\\ 1-2\left(\frac{1-P}{0.2}\right)^2, & 0.9<P<1\\ 1, & 1\le P\end{cases} \tag{5}$$

P ="tourist attraction" is determined using a hierarchical fuzzy inference system, which has four input variables and one output variable. Input variables characterize indicators of the attractiveness of certain types of recreation. For the territories of Zakarpattia Oblast, current types of rest and recreation can be grouped into four groups: p_1= "attractiveness of rest in catering establishments"; p_2="attractiveness of winter recreation"; p_3="the attractiveness of recreation in nature"; p_4="attractiveness of excursions and viewing of historical and cultural monuments".

The attractiveness indicator of the attractiveness of excursions and a review of historical and cultural monuments depends on two parameters:

- geographical coordinates of historical and cultural monuments and places of interest for tourists in the Carpathian region;
- rating assessments of the importance of the above-mentioned objects determined by experts. The indicator of attractiveness of excursions and viewing of historical and cultural monuments is determined by:

$$p_4{}' = \sum_{i=1}^{m}\left(\pi_i \times e^{-\frac{(N-\pi_i)r_{kl,i}^2}{\sigma^2}}\right), \tag{6}$$

where: $r_{kl,i}$ – the distance between the territory for which the indicator is calculated and the tourist and recreational object of historical and cultural significance; σ – the root mean square deviation determines the shape of the function, (a quantile of order ½ determines the “optimal distance” at which the indicator halves); π – rating evaluation of the recreational attractiveness of the historical and cultural object; N is the maximum value of the rating (when m=N, all visitors will visit the recreational facility). We will use it to assess the expediency of including one or another tourist object of historical and cultural significance in the route.

4. Project production and testing

Following the principles of React application development, the first thing to do is to highlight individual components (and sub-components) in the layout and give them names. Deciding whether to make an element a separate component usually uses the same approach as deciding whether to create a simple function or an entire object. The principle of single responsibility can be applied: each component must handle one task. If the functionality of a component increases over time, it should be broken down into smaller subcomponents.

It was decided to highlight such main components as:

- Header.
- Map (the main component of the map).
- Menu (functional menu).
- DrawerContent (a container for displaying the content of the application).
- PlaceDetails (registration of information about institutions or attractions).
- API (a component for asynchronous calls to external programming interfaces that store the necessary data).

In order to simplify the development of basic user interface elements for most sites, it was decided to connect and further use Material UI - a library of React components with open access to the code, which implements Material Design from Google. First of all, it was decided to create a page header, because at the initial stage of development, this element has the least complexity compared to others. Using the graphic components provided by the MaterialUI library, a basic version of the site header with a logo was created, thanks to the decision to use a specialized library for building user interface elements, it will be very easy to supplement this component in the future by adding such elements as search, menu, etc.

In the basic version, the cap is a blue rectangle with a container for internal content, but this is more than enough to get started. The next stage in the development of this web application was the process of creating a map component and implementing the GoogleMaps service. In order to connect the services provided by Google, you need to create an account in the Google developer panel and connect the necessary software products to your account, in this case it is:

- Google Maps API.
- Places API.
- Directions API.

According to the company's policy, the use of these services is paid and each individual API has its own price, the payment is deducted from the bank card connected to the payment account. Fortunately, you can use these services for free. Every month, Google provides developers with a certain amount of credits that are intended to pay for these services, that is, they can be used for free, as long as the amount of payment for the used services per month is not higher than the amount of free credits. As a result of working with the Google developer console, we receive a personal API key that will allow us to access any of the connected services, and it is also necessary so that the company that provides access to these services can control the use of their software products.

After completing the preparatory stages and receiving the API key, you can start working on one of the main components of this project - the map. For this purpose, it was decided to import the third-party specialized library google-map-react, which extends the capabilities of regular GoogleMaps. The main feature is the ability to use React components as markers on the map, while the standard GoogleMaps component accepts only images as markers. Without additional settings, this component will display only the base map without any points/markers/marks.

The GoogleMapReact component accepts a certain number of parameters, the list of used ones includes:

- API key that was obtained in the Google developer panel.
- Standard coordinates of the center of the map, which will be used in the absence of user geolocation.
- User coordinates collected at the time of connection to the website.
- Map zoom percentage to be used at the start of the map.

- Additional map options, such as design styles.
- The method that is performed in case of changing the coordinates of the central point on the map.
- The method that is executed when a child element of the map is clicked and will later pass its identifier as a parameter to other components, in this case the institution card acts as a child element.
- A method whose content is executed when receiving information from an external Google API.
- The yesIWantToUseGoogleMapApiInternals option, which allows you to use all the functions built into the GoogleMaps service.

As a result of the first stage of work with the map component, the component itself was created and a list of commands necessary for the generation of a base map with standard coordinates of the center in the city of Uzhhorod was formed.

At this stage, 2 components have already been created, but although they are fully functional, they are not displayed on the web page. The fact is that in React, with the basic settings, the main component App.tsx (or .jsx, the file format depends on the selected programming language, in this case – TypeScript) is responsible for displaying components on the page. Each created component, which is planned to be used in other files, must be exported using the export keyword and, accordingly, imported into the file where it is needed, import occurs using the import keyword.

After importing the necessary components into the environment of the App.tsx file, it is necessary to add their references in the form of child elements of the main block, which is returned as a result of the execution of the App method.

As a result of the work performed at this stage, a web page with a display of the created components was obtained, which has the following form: At the moment, no points are displayed on the map, for their generation various data about the institution are required, at least its spatial location in the form of an object with longitude and latitude.

Access to this API and many others can be obtained on the specialized service RapidApi, which is an incredibly large platform for developers who want to share their interfaces and, if possible, receive a monetary reward for it. Most of the APIs on this service are distributed according to the "freemium" policy - you can use this interface absolutely free of charge until the user's monthly quota does not exceed the limit set by the developer. If the user still exceeds the usage limit, his free access to the interface for this month will be blocked. If you wish, you can pay for an additional monthly request package, but for this educational project, free requests will be more than enough.

While the data from the Google API was received directly in the component, the data from the RapidApi service comes in the form of HTTP headers, and it is customary to use the axios library to work with them. To work with all APIs from the RapidApi service, it was decided to create a separate component.

The getPlacesData method is responsible for requests to the TravelAdvisor API. Requests are sent using a unique link, which is formed in such a way that when the type parameter is changed, the type of establishments that will be returned as a result of the request and recorded in the data variable will also change. Also, for identification in the request, you must specify your API key received in the RapidApi service. This API returns a list of institutions whose geolocation is within the specified limits, it was decided in our case that it will return institutions that are within the visible limits of the map. Due to the fact that there is no point in recording the coordinates of all four corner points, it was decided to use only two, namely the upper right and lower left points. Currently, the querying component has no connection to the map component, and the API cannot get the coordinates of the boundaries, and the getPlacesData method is never called. The central component of the App will act as a connecting link.

In React, if there is a need for a web page to visually display changes in the value of a certain parameter, it is not advisable to use ordinary variables of the let type, due to the fact that changing the parameter will not cause re-rendering of the component in which it is used and, accordingly, the changes will not be displayed. Hooks - special methods, one of which will allow you to solve the problem with rendering, will allow you to significantly expand the capabilities of the developer in React. We are talking about the UseState hook, which has 2 parameters - a variable to which data will be written and a method that will update the value of the variable and cause the component to be re-rendered.

At this stage, it was decided to create UseState hooks for all variables that will be needed in the main file in the future, they include:

- places – stores an array of establishments and entertainment;
- coordinates – stores the coordinates of the center;
- bounds – stores the coordinates of the boundaries;
- childClicked – stores the identifier of the child element of the map on which the user clicked;
- filteredPlaces – stores an array of establishments and entertainment sorted by rating on a 5-point scale;
- isLoading – used to record the state of receiving information from the API;
- type – stores the type of institutions selected by the user for filtering;
- rating – stores the limits of ratings of institutions selected by the user for filtering.

The useEffect hook is used to call the getPlacesData method, which takes two parameters:

- the first - executes the embedded code only when the page is first loaded;
- the second - accepts references to variables, the state of which must be monitored, if the value of one or more of these variables becomes different, then the program code stored in the first parameter will be executed again taking into account the changes.

This method will be called on the first run and every time the value of bounds and type changes, it will indicate that the data at the time of the call is not yet available and the component must be displayed with a visual loading display. Next, a check is made for the existence of a value in the bounds variable, if there is no value, the work will stop, if there is, it will continue. After that, the new bounds values (if any) are written to the bounds variable. The final stage is to call the getPlacesData method, pass it parameters necessary for operation, write the information received in response to the places variable and set a negative value to the state of the loading component, that is, disable its operation.

After a long preparatory process, all that remains before displaying the base points on the map is to transfer the necessary methods and variables from the App component to the Map component and implement the marker display functionality. The process of transferring components is quite simple and consists of packaging information and its further transfer to the required address.

At the initialization stage of the component, it is necessary to indicate which parameters it will accept, respectively, these should be the fields that were sent from the App component. Cases when data was sent and not received, or ready for reception but not sent, will lead to errors.

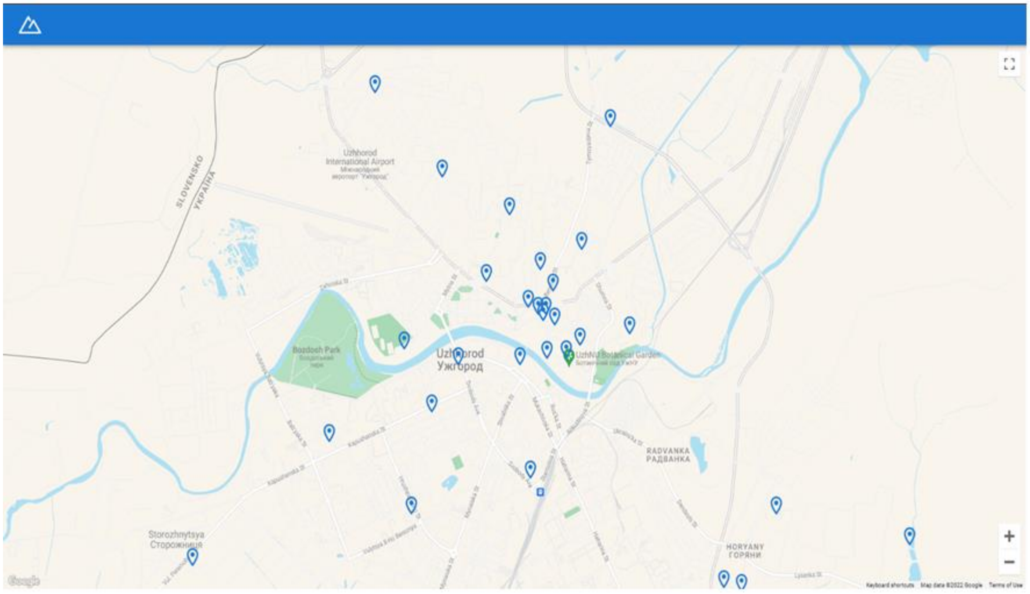


Figure 3: Displaying markers on the map

As a result of the manipulations, links to the necessary methods became available in the Map component and it became possible to record data with their help. For the API to work, map boundary data was required. For future functions, links and methods were created to monitor the state of the map and identify the clicked child element. All the necessary links were written in the corresponding parameters of the map, where onChange executes nesting at any change of its element and onChildClick, which is executed when clicking on a child element.

To display points on the map, it was decided to export the marker component as a separate file. To avoid errors and waste of program resources, a check is made to ensure that the places variable, which stores all the necessary information about establishments, is not empty.

In order to start displaying markers on the map, it is necessary to implement the basic functionality and marking components, it is quite simple to do. As a result of executing the Marker method, it is necessary to return an image that will be placed on the map according to the coordinates of the institution.

After all the work done, you will finally be able to see the location of all establishments on the map, hotels with any ratings will be displayed on the map based on the basic values of the filters. In order for the center of the map to not be tied exclusively to Uzhgorod at launch, it was decided to add reading of user coordinates. Obtaining user coordinates occurs with the help of a specialized function of browsers, namely `navigator.geolocation`. The useEffect hook was used to work with it. In the executable code of the first parameter, the function itself is called, and then the data obtained as a result of its execution is written into the coordinates variable using the setCoordinates method. Due to the fact that the array, which acts as the second parameter of the useEffect hook, is empty, the user coordinates are read only once - when the page is loaded.

If the user does not provide access to his geolocation, the web application will use the standard center, that is, the coordinates of the city of Uzhhorod. In order to give the user more opportunities to control the map, it was decided to start work on the search function, which will allow changing the center of the map by making a search request, and not only by moving the map manually. The ideal place for this kind of element is the site header. Google's Places API was used to perform the search. Autocomplete is a web service within the Places API that returns place predictions based on user input in response to an HTTP request. The query specifies the search text and optional geographic boundaries. The Service can be used to provide auto-complete functionality for text-based geographic searches, returning locations such as businesses, addresses, and points of interest as a user enters their query.

The content of the user input is recorded in the autocomplete variable, when one of the proposed search options is selected, the setCoordinates method is called and overwrites the coordinates of the cent.

As a result, a search field will be created, which responds to user queries and, based on them, offers options for places that the user could have tried to find. The city of Mukachevo was chosen as an example of input.

The last element that needs to be added to the site header is the weather display block, and it was decided to start working on this element. For its implementation, it is necessary to connect a specialized API provided by the OpenWeatherMap service[15].

This online service provides a paid (there is a functionally limited free version) API for accessing current weather data, forecasts and historical data. The source of data is official meteorological services, data from airport weather stations and private weather stations. The information is processed by OpenWeatherMap, after which, based on the data, weather forecasts and weather maps, such as cloud cover and precipitation maps, are built. The main idea of this service is the use of private weather stations, which help to improve the quality of the initial weather information and, as a result, the accuracy of weather forecasts.

The connection is made according to the same principle as with TravelAdvisor - in the file where requests are made to the above-mentioned API, a new getWeatherData method must be created. The content of this method is similar to the getPlacesData function, the main difference is the link on which the request is made and the necessary parameters. After creating the method, you need to add its call to the already existing useEffect hook, so the weather data will be updated every time the center coordinates change.

Next, the received information must be transferred to the Header component and, accordingly, the functionality of the weather display block must be implemented there.

In this block, a check is made to see if the weatherData variable is not empty, and only if the result of the check is true, this block is rendered. The temperature value is rounded to an integer using the Math.round function. The graphical representation of the weather is selected from the OpenWeatherMap database based on current readings.

Now the map shows the locations of hotels within the visible limits, you can see the weather status in the area and search by cities, institutions or other geographical points. But for convenient interaction with the application, the user will need a menu that can be used to manage the content of the map. Also, it is necessary to expand the range of information about the institution, not limited only to its location.

After finishing work with the cap, the web application looks like this:

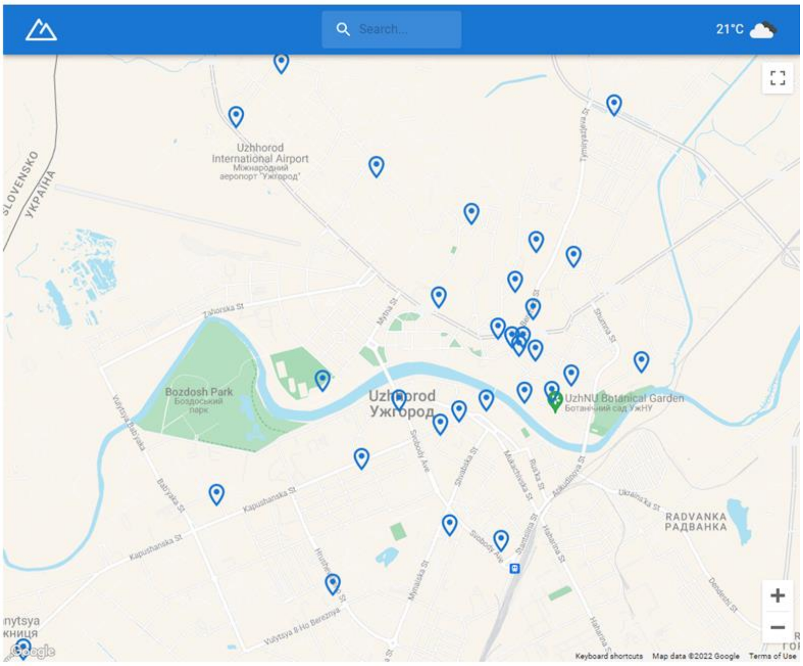


Figure 4: Visual representation of the component

The Hotel component is responsible for the svg image that will be displayed, when changing the name of this block, the corresponding image obtained from the MaterialUI library database will also change. When interacting with the button, the handleContentUpdate method is called, which will open the Drawer component, in which all content will be displayed. Drawer is a kind of "drawer" that moves out of the left part of the screen, it will display all the necessary content. The functionality of the method also includes setting the type of information that will be displayed in the Drawer block and on the map.

A check has been made that does not close the block if the next pressed button does not correspond to the type of data that is already displayed. That is, if the Drawer is open and the data type is set to “hotels”, then when you click on the “restaurants” button, the block will not be closed, but only its content will be reloaded. If the type for which the button is responsible matches the current type, the Drawer will be closed.

Inside the DrawerContent block, a layout for displaying information about institutions was created, the block title dynamically changes according to the value of the type parameter. An interface has been created for the filter block of establishments based on a 5-point scale. The filter takes four values:

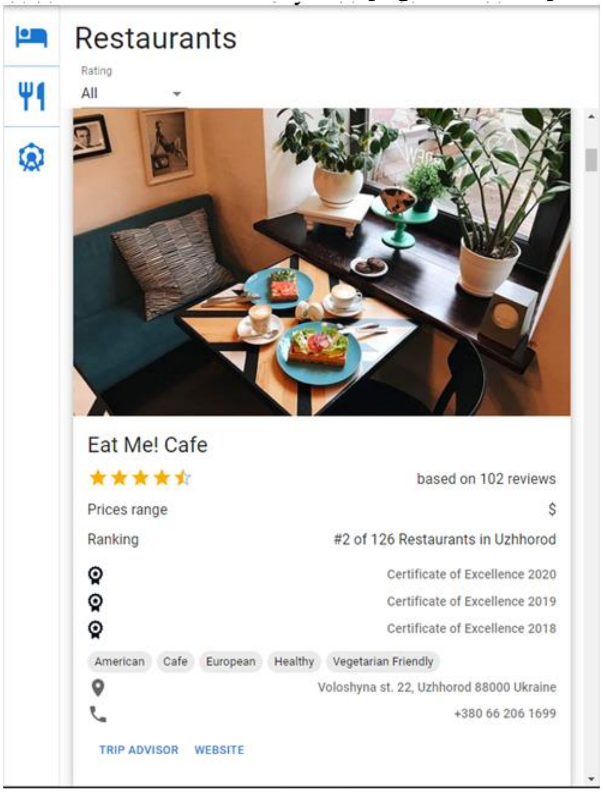
- Any assessment.
- Above 3 points.
- Above 4 points.
- Above 4 and a half points.

The content of institutions is organized in the form of a vertical table of cards. The data from the places variable is broken down using the JavaScript map method, and information about each individual institution is transferred to the PlaceDetails component, where a card is generated according to the appropriate template. Due to the fact that at the time the project was created, it was decided to use the TypeScript language, the syntax of which, unlike JavaScript, has strict data typing, it became necessary to create common interfaces for TravelAdvisor and OpenWeatherMap in which information about the types of fields obtained from the used APIs will be stored.

After describing the interfaces for all the fields that will be used to build the cards, you can start working on creating a template and displaying data. The first stage of creating a template will be to display an image of the institution, for this we check whether the object has a photo, if it does - we make a request to receive this photo in a large size, otherwise - we display a standard image, the link to which is stored in the file. env.

Further, with the help of appropriate containers for information, we display such information about the institution as:

- Title.
- Rating on a 5-point scale based on a certain number of reviews using the Rating component.
- Price range.
- Honors and awards.
- Categories of dishes served.
- Address.
- Phone number.
- Link to the institution's official website and its page on the TripAdvisor service.



Restaurants

Rating

All

Eat Me! Cafe

★★★★☆ based on 102 reviews

Prices range $

Ranking #2 of 126 Restaurants in Uzhhorod

Certificate of Excellence 2020

Certificate of Excellence 2019

Certificate of Excellence 2018

American Cafe European Healthy Vegetarian Friendly

Voloshyna st. 22, Uzhhorod 88000 Ukraine

+380 66 206 1699

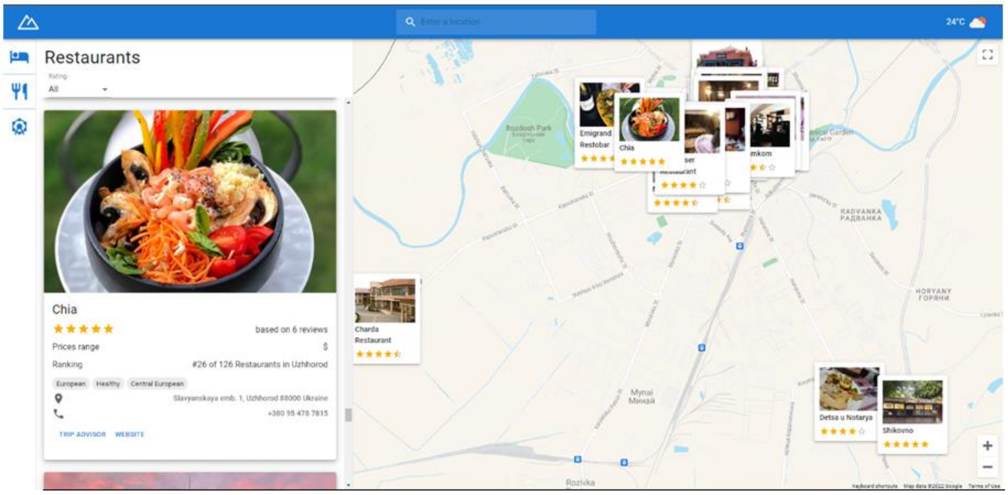
Figure 5: Visual representation of institution cards

Before displaying, the status of the fields is checked, if any of them are empty, they will not be displayed. This template works for all types of establishments whose data is obtained from the TravelAdvisor API. Institutions are filtered thanks to the previously created block in the DrawerContent component. Depending on the selected type of facilities, a request is made to the component in which requests to the API are made, thereby changing the link in such a way as to receive data on the required type of points of interest. Filtering by rating occurs thanks to an arranged

JavaScript method called filter, the result of which is an array of data that corresponds to the specified parameters, that is, the rating of the institution is taken and if it is higher than or equal to the specified in the filter, then we pass it to the PlaceDetails component.

The next stage will be to expand the informativeness of the marks on the map by creating your own card layout of establishments by analogy with the PlaceDetails component. Unlike the template presented in the PlaceDetails component, the Marker will only display an image, title, and rating. Based on the future possibility of adapting the site to other devices, devices with low resolution will display ordinary markers, if the screen of the device is larger, then the card template will be used. To do this, you need to add a new template in the Marker component, according to which the card will be generated.

Having fully utilized the data already transferred to the Marker component, we will be able to generate a certain number of establishment cards and place them on the map as markers. When hovering over a card, it moves to the foreground, avoiding the case when the cards will overlap each other without being able to see the contents of the lower elements.



Restaurants

Rating: All

Chia
★★★★★ based on 6 reviews
Prices range $
Ranking #26 of 126 Restaurants in Uzhhorod
European Healthy Central European
Slavyanskaya emb. 1, Uzhhorod 88000 Ukraine
+380 95 478 7815
TRIP ADVISOR WEBSITE

Figure 6: Visual representation of the result of the work performed

For the case when the user finds an institution that interests him on the map and wants to get detailed information about it, a function was created that will automatically scroll through the list to the institution that was clicked on. The last element that needs to be developed is the function of building routes. To implement it, you need to add two new buttons to the menu:

1. It is responsible for the formation of the route and records the point data in the IndexedDB database with a special name.

2. Builds a route on the map.

In order to add points to the database, you must also add a new button to the cards of institutions, after clicking on the button, the point will be recorded in the context of the application.

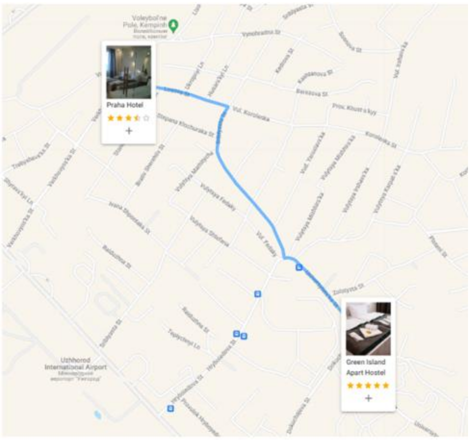


Figure 7: Route planning

After adding at least two points to the context, it is possible to save the route in the database. If the route exists, it will begin to be displayed in the content of the block with the formation of routes. When you select one of the generated routes and click on the corresponding button, the generated route will begin to be displayed on the map, along with the points related to it.

5. Conclusions

The research is based on the issue of using modern information solutions to improve user experience in the tourism trip planning. A web application for planning local tourist routes in Transcarpathia was also implemented. The developed product is an example of combining the technologies of expert evaluation, route optimization and working with public APIs and map services to create web applications. During the development of the project, the most popular tourist routes of Transcarpathia and the possibilities of graphic display of food and accommodation facilities on the map were studied, the React library was used using fuzzy modeling to determine the priority points of interest when laying the route.

6. References

[1] Y.Yang, J. L. Stienmetz, Big data and tourism planning. Information Technology & Tourism, 20(1-4), 2018, pp. 189–190. doi:10.1007/s40558-018-0127-6

[2] M. Cheng, Media Discourse on Big Data and Tourism Attractions in China. Perspectives on Asian Tourism. 2018, pp 215–229. doi:10.1007/978-981-10-8426-3_13

[3] T. Gajdošík, Big Data Analytics in Smart Tourism Destinations. A New Tool for Destination Management Organizations? WissenKompakt Medizin, 2019, pp. 15–33. doi:10.1007/978-3-030-03910-3_2

[4] M.Tao, M. Z. Nawaz, S. Nawaz, A. H. Butt & H. Ahmad, Users' acceptance of innovative mobile hotel booking trends: UK vs. PRC. Information Technology & Tourism. 2018, doi:10.1007/s40558-018-0123-x

[5] M.Scaglione, C. Johnson & P. Favre, As time goes by: last minute momentum booking and the planned vacation process. Information Technology & Tourism, 21(1), 2018, pp. 9–22. doi:10.1007/s40558-018-0133-8

[6] F. Ricci, L. Rokach, B.Shapira, Recommender Systems Handbook: Second Edition. Springer Science+Business Media New York, 2015, p. 1003. – ISBN 978-1-4899-7636-9

[7] A. Delic, J. Neidhardt, T. N. Nguyen & F. Ricci, An observational user study for group recommender systems in the tourism domain. Information Technology & Tourism, 19(1-4), 2018, pp. 87–116. doi:10.1007/s40558-018-0106-y

[8] Y. Shang, C. Wen, Y. Bai and D. Hou, "A Novel Framework for Exploring the Spatial Characteristics of Leisure Tourism Using Multisource Data: A Case Study of Qingdao, China," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 6259-6271, 2022, doi: 10.1109/JSTARS.2022.3196002.

[9] O. Artemenko, V. Pasichnyk and N. Kunanets, "Using Mobile Location-based Recommender Systems for Providing Real Time Recommendations for Social Distancing Urban Route Planning," 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 2020, pp. 305-308, doi: 10.1109/CSIT49958.2020.9321969.

[10] Y. Zhu, "Introducing Google Chart Tools and Google Maps API in Data Visualization Courses," in IEEE Computer Graphics and Applications, vol. 32, no. 6, pp. 6-9, Nov.-Dec. 2012, doi: 10.1109/MCG.2012.114.

[11] I. P. G. A. A. Putra, E. Sediyono and A. Setiawan, "E-land design of mobile application for land information system using Android-based Google Maps API V2," 2017 International Conference on Innovative and Creative Information Technology (ICITech), Salatiga, Indonesia, 2017, pp. 1-5, doi: 10.1109/INNOCIT.2017.8319145.

[12] D. Bura, M. Singh and P. Nandal, "Predicting Secure and Safe Route for Women using Google Maps," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel

Computing (COMITCon), Faridabad, India, 2019, pp. 103-108, doi: 10.1109/COMITCon.2019.8862173.

[13] P. Cheng, J. Hu, Z. Yang, Y. Shu and J. Chen, "Utilization-Aware Trip Advisor in Bike-Sharing Systems Based on User Behavior Analysis," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 9, pp. 1822-1835, 1 Sept. 2019, doi: 10.1109/TKDE.2018.2867197.

[14] B. A. Pramajuri, A. Gormantara, E. Widarti and A. J. Santoso, "Recommendations for Tourism Sites Using the Mamdani Fuzzy Logic Method and Floyd Warshall Algorithm (Case Study in Yogyakarta)," 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019, pp. 590-595, doi: 10.1109/ICOIACT46704.2019.8938592.

[15] C. Trattner, A. Oberegger, L. Marinho & D. Parra, Investigating the utility of the weather context for point of interest recommendations. Information Technology & Tourism, 19(1-4), 2018, pp. 117–150. doi:10.1007/s40558-017-0100-9