IIT PALAKKAD

ID1110 - Introduction to Programming
Course Project

Team members

- 112301024 : Raagam Parmar Hitesh
- 122301034 : Rajdeep
- 122301038 : Souransu Roy

Event Mapper

Introduction

Background and Context

Indian Institutes of Technology or IITs are premier Indian institutions aimed at training scientists and engineers to develop a skilled workforce to support the nation's development. IIT Palakkad, being one of them, sees an influx of outsiders at various times of the year.

The visitors, being new to the campus, face difficulty navigating it and keeping a track of the ongoing events. Our app aims to make this process more convenient, both for the organisers of events and the visitors.

Problem Statement

Currently, there is no convenient way to keep track of the events going on in the various buildings of IIT Palakkad. People who visit the institute face difficulties in navigating the campus. Sometimes, students miss out on events due to the lack of a localised platform for managing all events.

Objectives

- Developing an application to enable visitors to navigate through the campus with ease.

- Developing a localised platform for both visitors and organisers to explore and manage events via the application.
- Adding features to make the usage of the application convenient and user friendly, and making the application scalable.

Solution

Our application provides users with a map of the campus and allows them to interact with it. Various landmarks are marked out on the map and the user can click on a landmark to see a list of events at that venue. A concise description of each event, as defined by the organiser, is displayed with the event. Users who are signed in as organisers can add new events, with details such as the event's timings, a short description, etc.

We make it easier for visitors to navigate through the campus with the map, and it allows them to plan their visit as per the events displayed. Furthermore, organisers can directly interact with the app by adding and removing events. This makes our application a localised platform for both visitors and organisers to explore and manage all events, respectively.

Features such a convenient and secure log-in system and warning organisers of possible time clashes with other events make the application user friendly.

Significance and Motivation

We faced a considerable amount of difficulty during Petrichor '24 in navigating through the events that were taking place throughout the campus. The events were difficult to keep track of, due to them being large in number. As a result, we missed out on some really interesting

events and workshops. Petrichor ‘24 used an application called Tangle to attempt to manage all events. Navigating through the application was cumbersome and it had no map incorporation. This made Tangle extremely inefficient and ineffective.

After experiencing this, we decided to make our own application to fix the issues that we encountered, and to provide additional features.

Our application will prove to be extremely significant, both in our day-to-day life in college - for events that the clubs regularly conduct, and in campus-wide events such as Petrichor - for the events and workshops held.

Project Overview

Project Goals and Scope

The primary goal of the project is to provide a localised platform for exploring and managing events going on in the multiple venues in IIT Palakkad. We aim to provide convenient and user-friendly methods for the organisers to manage the events, along with features such as warning organisers when their newly defined event clashes with another event, etc. Visitors and organisers should be able to interact with the map in a similar manner, with added features for organisers. Another important aspect of our goal is to make the entire project scalable. For example, the act of updating the map to incorporate more of the campus should be achieved easily.

The current scope of the project is the Nila Campus, but we aim to expand it to the Sahyadri Campus as well.

Project Timeline

- 26/02 - 29/02 : Initial ideas were proposed and discussed
- 04/03 - 10/03 : Event Mapper was finalised and details were worked out
- 11/03 - 17/03 : Various libraries and modules were explored
- 18/03 - 24/03 : The Kivy library was decided on, and basic features of the app were discussed in further detail
- 26/03 - 07/04 : All members studied the Kivy library and divided the tasks among themselves
- 08/04 - 23/04 : The individual portions were formed.
- 24/04 - 27/04 : The individual portions were compiled. Some new features were introduced as well.

Project Repository (GitHub)

Team Members and Contributions

I. Raagam Hitesh Parmar
   - Designed the campus map in AutoCAD, with Google's satellite imagery as reference.
   - Designed the map page in the application that handles interactions with the map.

II. Rajdeep
   - Designed the starting page and the login page, which manages the credentials database.
   - Designed the functionality for adding and removing events, which manages the events database.
   - Worked on the formatting of buttons, labels, etc.
   - Compiled the individual portions.

III. Souransu Roy
- Designed the visual aspect of landmark buttons and the displayed events.
- Designed a base framework for storing and managing events.

Methodology

Approach and methodology employed

We started by exploring various libraries that we could make use of for developing our application. There were three promising candidates - Kivy, Tkinter and PyQt. We decided on Kivy being the best candidate, due to it being cross-platform and reactive. PyQt provided these features as well, but we chose Kivy due to it being better documented, and due to the wide range of resources available online to get started with it.

As Kivy revolves around object oriented programming (OOP), we decided to strengthen our concepts of OOP before moving on to actually using the library. We watched online tutorials and went through Kivy's documentation whenever necessary to get well acquainted with the fundamentals of the library.

Once we felt that we knew enough about the basics of Kivy, we moved on to learning about the specifics that we would need to employ for our program. We had to refer to the documentation frequently for this, and our decision to choose Kivy over PyQt proved to be a good one.

Kivy has a sub-language called kvlang to make it easier to define the visual aspects of an application, but as our application was going to

be logic-heavy, we decided on not using kvlang at all, due to the difficulty in combining kvlang with a logic-handling Python file. The documentation of most Kivy objects demonstrates the code both with and without kvlang, which again demonstrated the helpfulness of the documentation.

After getting extremely well acquainted with Kivy, we separately developed the starting and login pages, the map page and functionality for the buttons on the map page. Then, we compiled all of the code. Special care was taken to follow the PEP-8 guidelines and to add comments in the code whenever necessary.

After the individual portions were ready, we combined them into one single Python file. We fixed some bugs that had surfaced and worked on some additional features to enhance the user experience. Then, we simulated our application with actual events taking place in our campus.

Tools, Technologies and Frameworks Used

Our project is completely Python-based.

We worked with the library, Kivy. It allows for the development of cross-platform graphical user interface applications. Due to Kivy's nature, the framework that our project is built upon is Object Oriented Programming.

To manage the text files that act as databases, we used Python's text file handling facilities.

To encrypt passwords before storing them in the credentials database, we used the SHA512 algorithm from Python's hashlib module.

The functools module was used to efficiently enable the functioning of landmark buttons in the map page. Using functools made our code more scalable as well, by providing ease in defining new venues or removing existing ones.

To manage the date and time aspect of events, Python's datetime module was used.

The following Kivy objects were used extensively in the program :

- Label
- Button
- TextInput
- Image
- LabelBase
- Screen
- ScreenManager
- FloatLayout
- ScatterLayout
- GridLayout
- ScrollView
- Popup

Experimental Setup and Data Collection Methods

Throughout the stages of our project, we have used actual events going on in the campus to simulate the functioning. For the sake of realism, we gave accounts to each of the major clubs and defined events for them. We used this data to test our features, such as event timing clash detection, sorting events based on their timing, filtering them out based on their timing, etc.

This gave us valuable insights into how the project would look once it's completed and scaled, and gave us ideas for new features to add.

Results and Analysis

Challenges Faced

We faced several challenges during our project. A prominent one at those has to do with database management. None of us are familiar with SQL, and to use Python's SQLite or MySQL modules, one needs to give the program SQL commands. Without using a database, we would be unable to demonstrate the project across multiple platforms.

Another significant challenge was not being able to implement live location tracking through our app. We had initially planned on displaying the user's current location on their map, to allow them to navigate the campus easier, but we were unable to do so, mainly due to time constraints.

The map of our campus that Google Maps' satellite view displays is extremely outdated. On top of that, it's of extremely low resolution. This resulted in us having to scrap our idea of incorporating Google Maps in our project.

Solutions to the Challenges

To work around the problem with database management, we decided to use a text file to simulate a local database. As Python has in-built file handling capabilities, we were able to work with our text-database extremely well, which simulated and demonstrated how an actual database would be managed.

To work around the low resolution of Google Maps' satellite images, we decided to use them as reference and made our own map using

AutoCAD. This allowed us to have a high resolution image of the campus, and gave us the freedom to choose our own colour scheme.

Conclusion and future work

Assessment of project success

As per what we had planned, we consider our project to be successful. We were able to work around well for most of the challenges that we faced, which didn't result in a significant quality difference in the project. We ended up learning more about Python due to the workarounds that we had to come up with, which we see as a positive point.

Lessons Learned

We learned a great deal about Python and programming, in general, from our work on this project. Here are a few significant points :

- Python's versatile collection of modules and libraries can come in very handy during large projects.
- Even with the wide range of libraries that Python has, it's still very challenging to implement certain things.
- Following a convention for writing code (in our case, PEP-8) is extremely useful for collaborative projects.
- Detailed planning before actually working on a project can go a long way.
- A well documented library is much more useful in the longer run than its counterpart, and should always be preferred.

- Different programming languages specialise in different domains, and they should be used accordingly.

Recommendations for Future Improvements

There are some useful improvements that we can make to our project to enable us to scale it further.

- The map can be expanded to include the Sahyadri campus as well.
- Database management using SQLite or MySQL can be used to enable cross-platform functioning of the program.
- Visitors can have credentials as well, which will allow them to register for events and organisers can have a good idea of how many attendees they should expect. This will allow them to prepare well for their events.
- We can make use of app notifications to remind users of certain events.

Team Members' Github accounts

1. Raagam Parmar Hitesh
2. Rajdeep
3. Souransu Roy

Resources

- Kivy documentation
- hashlib documentation
- datetime documentation
- functools documentation