International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-14 Issue-9, August 2025

Cross-Platform Development of a University Campus Navigation System Using Kotlin Multiplatform

Prathamesh Bhusekar, Shivanand Manyar, Hrishita Thanekar, Vidya Chitre

***Abstract**: Navigating vast university campuses is usually a significant problem, especially for newcomers or fresh admissions. Traditional resources, such as static maps or markers, often fail to provide an efficient method for guiding users through complex and multi-level campus areas, particularly within indoor environments. This document presents a cross-platform mobile app that has been implemented using Kotlin Multiplatform to address this urgent problem. The proposed solution is a campus navigation app that is entirely independent of internet connections and GPS, thereby being highly efficient in areas where connections or GPS signals are weak, such as building interiors. The core idea is to display users a series of preloaded checkpoint images and corresponding text prompts, rather than relying on dynamic GPS location tracking. The checkpoint-based navigation allows users to navigate around the campus manually advancing through sequential visual cues, thus providing user control and simplicity. The development is carried out using Kotlin Multiplatform to enable the sharing of logic across Android and iOS, thus providing instant development and identical functionality across both platforms. Shared modules are used for navigation logic, route control, and user input, while platform-specific code is reserved for rendering the user interface only. The application design focuses on simplicity and a visual direction system that maximises accessibility and clarity. This research work was conducted to fill the gaps in indoor navigation systems and present a comprehensive solution for educational campuses where real-time GPS technology is not economically viable. The value of this research lies in its platform independence, offline capability, and focus on practical implementation in real-world environments. By presenting a feasible solution to navigating campus spaces without GPS and internet connectivity, this research work contributes to software engineering, human-computer interaction, and location-based applications. Future development could include the integration of augmented reality and indoor positioning technologies to improve the overall user experience.*

Manuscript received on 21 June 2025 | First Revised Manuscript received on 28 June 2025 | Second Revised Manuscript received on 17 July 2025 | Manuscript Accepted on 15 August 2025 | Manuscript published on 30 August 2025.

*Correspondence Author(s)

Mr. Prathamesh Ravindra Bhusekar*, Department of Information Technology, Vidyalankar Institute Of Technology, Vashi (Maharashtra), India. Email ID: prathambhusekar@gmail.com, ORCID ID: 0009-0007-5759-8727

Mr. Shivanand Manyar, Department of Information Technology, Vidyalankar Institute Of Technology, Vashi (Maharashtra), India. Email ID: shivanandmanyar@gmail.com, ORCID ID: 0009-0007-2819-5451

Ms. Hrishita Thanekar, Department of Information Technology, Vidyalankar Institute of Technology, Vashi (Maharashtra), India. Email ID: hrishitat814@gmail.com, ORCID ID: 0009-0005-6795-0128

Dr. Vidya Chitre, Department of Information Technology, Vidyalankar Institute Of Technology, Vashi (Maharashtra), India. Email ID: Vidyadchitre2k14@gmail.com, ORCID ID: 0000-0001-8528-1048

***Keywords**: Campus Navigation, Kotlin Multiplatform, Cross-Platform Development, Mobile Application.*

Abbreviations:
AR: Augmented Reality

I. INTRODUCTION

Setting up campuses in new, rapidly evolving modern academic environments makes the campus larger and more complex in terms of its size, structure, floors, layout, and infrastructure. While infrastructures for universities have increased to meet their rising enrollment levels and diverse programs, navigating them has become an increasingly challenging task. New students, staff, and visitors often become confused by the lack of clear information, which can heighten confusion, lead to delays, and result in a poorer campus experience. This is particularly evident in the first weeks of an academic year or during events that attract outsiders, such as conferences or festivals. Traditional navigation aids such as static maps and signboards mounted at various points on campus offer little help.

These applications provide views of the campus, but most of them cannot be used in real-time or indoors, where many destinations, such as classrooms, labs, and administration offices, are located. Other navigation applications, such as Google Maps, though quite helpful outdoors, lack advanced indoor capabilities to guide users through specific buildings or even individual floors. There has been an increasingly felt need to devise a more personalised and appropriate way of navigating the campus. The problem this research paper attempts to address is: How Vidyalankar Institute of Technology freshmen struggle to navigate the overwhelming and unwieldy infrastructure that the institute possesses.

It will be challenging for students new to campus life and visitors to keep track of their classrooms, labs, or any other facilities in the early days of campus life. Misorientation tends to lead to such ineffectiveness, which ultimately affects the smooth running of campus life and reduces user satisfaction. People become so addicted to their smartphones and various mobile applications for day-to-day activities, instead of just having real-time updates of indoor and outdoor navigation at an unprecedented level of personalisation related to campus geography. There needs to be something more dynamic. This work will develop a cross-platform mobile navigation application using Kotlin Multiplatform to get an effective cross-platform solution

Kotlin Multiplatform will also enable developers to share code in a single codebase across these platforms, in

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

addition to supporting the features of some of these platforms when necessary. In this, the proposed application will utilise the platform to guide users around Vidyalankar Institute of Technology with real-time and precise location information, both indoors and outdoors. This proposed application will be very user-friendly; hence, navigating around the campus will be easy, especially for first-time visitors. Originality: The work employs a unique approach to cross-platform mobile application development using Kotlin Multiplatform.

Unlike other traditional approaches, most of the development needs to be implemented separately for each platform. Kotlin multi-platform enables great code sharing across platforms, which is already noted to save money and time in the development process. Efficient development ensures that the application is compatible across a wide range of devices, maintaining quality and uniform performance while retaining functionality that meets the specific requirements of each platform. Additionally, the work opens the door for further development in future years, likely bringing added value to the user experience, which includes indoor positioning systems and augmented reality. It therefore calls for appropriate navigation tools, as demand continues to increase and is one of the significant problems that this research aims to address. This research study aims to summarize the overall improvement of students' and visitors' experiences on campus by providing a means of developing a cross-platform compatible campus navigation application using Kotlin Multiplatform, which will increase ease of accessibility, efficiency, and the general feeling of navigating the institution's space.

II. RELATED WORKS

Many recent research activities on campus navigation have approached the topic from various angles. The themes of these studies encompass the user experience and how it can be engineered using sophisticated technologies. The studies primarily involve GPS-based systems, context-aware navigation, indoor positioning, and event-driven applications, whose goals are to provide appropriate and accurate guidance within a campus setting.

Gryzun, Shcherbakov, and Bida proposed the design of an information system to improve wayfinding in and around university campuses [1]. The research combines geospatial information, campus building facilities, and pathfinding algorithms to enable students and guests to identify buildings, classrooms, and offices effectively. Building floor plans, QR-code markers, and mobile interfaces are utilised by the system to enable wayfinding, mirroring the complexity of multi-level path mapping and the need for real-time information updates. Unlike solutions based exclusively on GPS technology, the system compensates for indoor limitations employing image recognition and coded reference points, which mirrors the idea of offline navigation designed explicitly for campus environments. The research is a valuable contribution to the construction of location-aware applications in the academic context by addressing usability and accessibility concerns of practical interest.

The paper describes an application for indoor navigation on campus that provides real-time directions using Wi-Fi and sensor information to improve the positional accuracy in university buildings [2]. Abbas Helmi and others emphasised the need for precision in indoor environments, where GPS signals are typically weak. The system is compatible with Android and utilises visual markers, along with floor plan information, to provide effective navigation for users. The paper presents a comparative analysis of sensor-based tracking methods and discusses the challenges associated with location calibration. The application offers an enhanced user experience by emphasising convenience, accessibility, and space awareness, particularly for first-time users. This aligns with the objective of our study in indoor navigation. However, our project differs in that it operates completely offline, utilising visual and Kotlin Multiplatform to achieve dual-platform compatibility.

Ang and others developed a campus guide system based on Google Glass, intended to be a hands-free, real-time navigational aid for students and visitors [3]. The system relies on augmented reality (AR) overlays combined with GPS for walking directions, with wearable technology use in mind. It utilises voice commands and location tracking to significantly enhance the user experience. However, despite being innovative, the solution's effectiveness relies on constant internet connectivity and compatible hardware, which limits its scalability. Our app, developed in Kotlin, on the other hand, eschews the use of GPS or other devices by employing image-based navigation through generated checkpoints. However, this effort highlights the increasing trend towards innovative and accessible solutions for campus mobility that aim to reduce user effort.

Sparsh, Jadaun, and others proposed an offline campus navigation system that does not require internet connectivity or GPS [4]. Their app uses static maps and floor plans to provide navigational instructions through interactive buttons and text-based commands. Users navigate manually through buildings by choosing predefined checkpoints, a process that closely resembles the structure of our project. This study reinforces the notion that efficient indoor navigation may not rely on real-time connectivity. The study mentions minimized battery usage and app responsiveness in offline mode, aspects that are relevant to our design goals. The standard approach highlights the viability of offline navigation apps, warranting our application of Kotlin Multiplatform to implement this feature across platforms.

Anpat and others. Proposed an Android-based solution in [5] that also acts as campus navigation. The system, similar to the one discussed above, is based on GPS for outdoor route planning and guidance. The system targets freshers and visitors, enabling them to easily find facilities on any university campus. Still, it does not differ significantly from previous approaches to provide proper indoor navigation, highlighting the greater drawback of GPS-based campus navigation systems [6].

Systems that single-mindedly focus on solving the indoor challenge are indoor navigation systems. The system can provide location information even when GPS is not accessible, making it suitable for large campus buildings or multi-story complexes [7]. This solution is highly effective in environments where GPS is unavailable, but it requires the deployment of additional hardware throughout the

Retrieval Number: 100.1/ijitee.H111614080725
DOI: 10.35940/ijitee.H1116.14090825
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

campus in the form of beacons.

Another area of interest in the campus navigation system is incorporating event-driven navigation. For example, the transportation system changes routes and adverts based on live campus events such as lectures, workshops, or even parties. This event-action-based navigation system provides users with current information. It directs them to places where various activities occur, ensuring they are always informed about events happening on campus and reach their desired destination as promptly as possible. Such systems are beneficial for dynamic campus environments with frequent events and building changes. However, continuous internet connectivity is often required in these systems to push real-time updates, which can limit their functionality in environments with limited or no internet connectivity.

Cross-platform development has emerged as a very critical factor in enhancing the user experience [8]. While most campus navigation systems focus solely on Android [9], some authors have developed cross-platform applications. Most of the above systems are created exclusively for Android, leaving iOS users out. For this project, a Campus Navigation App was designed using Kotlin Multiplatform, a framework that enables code sharing across Android and iOS platforms. Then, the app will provide a consistent experience across Android and iOS, and most importantly, it will require reduced effort in maintaining two separate codebases. Due to the handling of business and navigation logic across platforms within Kotlin Multiplatform, users will receive a consistent high-class navigation experience, regardless of whether they are navigating through their device.

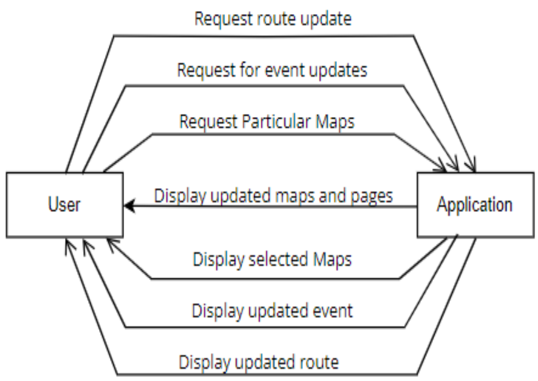
The above project is even more unique because it focuses on offline navigation using locally stored checkpoints. This is also the case with offline research conducted by Chotbenjamaporn and others. [10], An offline web-based system is used to demonstrate how offline functionality can be a requirement for the continuation of the service without interruption, even in the presence of an unreliable internet connection. The system does not rely on dynamic updates or real-time location tracking through GPS or beacons, but rather on predefined routes and checkpoints that guide users through static images and text. It is helpful because it allows one to navigate the campus without needing GPS or constant Internet connectivity. This proves to be quite useful in environments where connectivity is limited, such as those campuses located in rural areas and buildings that require extensive shielding.

In conclusion, although GPS and context-aware systems have dominated solutions for campus navigation over the last few years, recent advances in indoor positioning and cross-platform development provide exciting alternatives [11]. Our Campus Navigation App fills all the gaps in these systems with a comprehensive offline solution through checkpoint-based navigation. Furthermore, it has considerable scope for future enhancements, allowing the user experience to be augmented with AR. Additionally, Bluetooth beacons can be integrated indoors, further improving the accuracy of the navigation system in such areas [12]. Kotlin's official documentation supports this development process by providing libraries and tools for building scalable multiplatform applications [13]. The effectiveness of Kotlin-based cross-platform approaches has also been validated through recent comparative studies in mobile development frameworks [14].

III. PROPOSED SYSTEM

The proposed university campus navigation guidance and maps application will function on both iOS and Android phones, without requiring internet access or GPS. It will allow users to use it anywhere, even when the system is out of range. We have implemented the app for the main campus building. It can be used by faculty members, university students, staff, invigilators, parents of students, and university visitors. Figure 1 shows the Level-0 Data Flow Diagram for the application.

This flow diagram illustrates how the user interacts with the system to obtain a map en-route to their final destination within the application for campus navigation, developed using Kotlin Multiplatform. Users interact with the application by requesting updates on their navigation routes, which may have changed due to environmental or campus information updates, as well as specific maps for different campuses, such as buildings or floors. This application then refreshes and presents the most recent maps or layouts of the campus, maps related to the user's inquiry, and information regarding the latest update, along with any new route changes. This flow highlights core functionality—data exchange, as users can easily orient themselves on campus. Kotlin Multiplatform ensures that this model of interaction is consistently reproduced on both Android and iOS, leading to the highest possible development productivity and the capacity to build a dynamic application that solves a wide range of problems in a complex university setting.



[Fig.1: DFD Level 0]

IV. IMPLEMENTATION

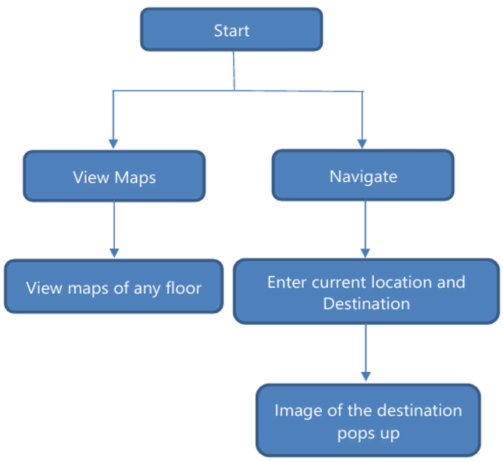
The Campus Navigation App was developed using Kotlin Multiplatform, enabling the sharing of effective code across Android and iOS, while also supporting platform-specific applications where necessary. It's an entirely offline app that doesn't require GPS or an internet connection. Instead, navigation is accomplished through a series of checkpoints, represented by images and text. Those checkpoints then guide the user through a series of steps from location to

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

location, instructing them on the distance to travel and the general direction, but leaving the actual execution of the command to the user based on their current environment.

The core module is the epicentre of the application and where the logic related to the actual application's navigation and implemented checkpoints resides. This straightforward logic does everything: it processes user input, remembers which checkpoints the user has already traversed, and informs them of the direction to the nearest future checkpoint. A usual navigation session guides the user via a list of checkpoints.

Each of them would be the picture of something—a building, or some other landmark—but would include small text with instructions, such as "Walk 100 meters straight, and at the library, turn left." As this application does not provide GPS tracking of the location, the user will need to locate these checkpoints themselves by following the given directions. That is, Kotlin Multiplatform should be applied to share core logic between Android and iOS, ensuring consistency in the user experience across both platforms. "Actual" would be used in places where functions differ in their implementation from one another. However, in this case, as the app is designed to work without needing platform-dependent location services and without online features, this is mostly just shared logic. This is because the platform-dependent code is primarily responsible for natively rendering the user interface on Android and iOS. In contrast, logic related to routes and checkpoints resides in the shared module.



[Fig.2: Architecture of the Application]

To graphically describe the user navigation flow within the application, Figure 2 illustrates the entire navigation process. As shown in the flowchart, when the application is launched, users have two main options: "View Maps" and "Navigate." If the user chooses "View Maps," they are provided with access to floor maps for any floor of the campus building. However, if the user selects "Navigate," they are asked to provide their current location and destination. Then they are presented with a series of images and text instructions to guide them through the navigation process. The navigation continues through the intentional selection of checkpoint images, thus providing an entirely offline and GPS-independent experience, which is particularly helpful in a university campus setting.

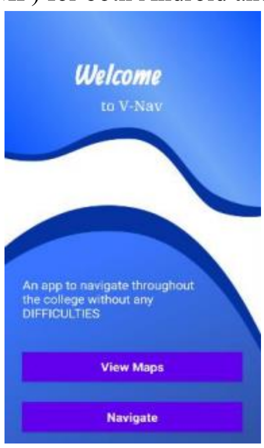
This is due to the challenges of maintaining a user-friendly navigation system throughout the development process without real-time location tracking. In other words, the user will not have access to GPS; therefore, they must rely on visual and textual hints displayed at every given checkpoint.

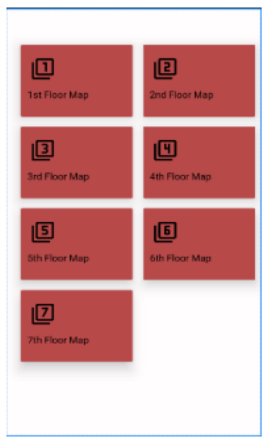
In this regard, the app focuses on providing lucid and easy-to-understand directions with highly recognisable images, enabling users to orient themselves better and feel more confident in following routes. Another such design-related feature is the simplicity of the user interface, which focuses on only the two most relevant pieces of information. Key features include the Checkpoint Navigation Flow, which guides users across campus via a series of waypoints, and map interaction, allowing users to view a simplified layout of the campus and select a destination. All the pictures of the checkpoints and the text instructions are stored in the device itself, making the application lightweight and quick to use in offline mode. The Campus Navigation App is developed cross-platform with native performance and features for both Android and iOS versions, utilising Kotlin Multiplatform for user experience. These are important subjects to consider when evaluating what the system has been able to deploy as a very effective and easy way of navigation. At the same time, locally based checkpoints ensure accessibility in areas with network unavailability or a lack of GPS and Internet connectivity.

V. RESULTS

The Campus Navigation app benefits from Kotlin Multiplatform's ability to share business logic—such as routing, event updates, and data management—across both Android and iOS. Libraries like Kotlin x Serialization enable consistent data parsing, storage, and networking across platforms without requiring platform-specific code. The shared module handles most of the app's logic, while platform-specific code is used for device-dependent functionality, such as rendering the UI.

The screenshots provided depict various key activities within our Campus Navigation app, along with the details for each, and their potential implementation based on the structure, as outlined below. Utilising Kotlin Multiplatform (KMP) for both Android and iOS.





[Fig.3: Introduction Screen] [Fig.4: View Maps Activity]

The Introduction Activity (Figure 3) features a simple interface screen that allows

Retrieval Number: 100.1/ijitee.H111614080725
DOI: 10.35940/ijitee.H1116.14090825
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

users to select their desired option, either navigating to their destination or viewing the campus floor maps. The View Maps Activity (Figure 4) displays a simple interface that allows users to choose from various floor maps (e.g., 1st Floor Map, 2nd Floor Map, etc.). A corresponding link to a map of that level in the building is associated with each button.

The Activity for Showing Maps (Figure 5) displays a campus map or a floor plan with marked points of interest, likely allowing users to see their route in relation to different checkpoints or rooms on that floor. The Navigate Activity (Figure 6) features input fields where users can enter their current location and their desired destination, followed by a prominent "Navigate" button. Once users provide this input, the app begins calculating the route between checkpoints. This navigation logic, including the routing algorithm and checkpoint management, is part of the shared codebase, ensuring identical performance on both platforms. The visual elements, such as the input fields and buttons, would be handled using platform-native UI frameworks.

[Fig.5: Activity for Showing Maps]

[Fig.6: Navigate Activity]

[Fig.7: Activity Showing Destination Images and Directions]

The Activity for Navigation (Figure 7) is a step-by-step navigation guide provided within the app. This includes textual guidance, along with additional images of a real-world scenario—for example, "Go right from A Block, walk about 25m." It navigates users to their target using a checkpoint-based navigation mechanism, which operates without GPS and instead relies on a set of predefined images, followed by manual clicking of a "Next" button. If implemented, this would serve as the central logic for fetching these images and instructions, forming part of the shared module. These images and instructions will be rendered for the user by platform-specific UI components. Put simply, Kotlin multiplatform code maintains the sequence of navigation steps and guides a user through a course.

Each of these activities shares logics for navigation, handling maps, and processing user input. At the same time, native UI components delivered by the platforms develop a native experience on Android and iOS. The app will be based on Kotlin Multiplatform, thereby ensuring consistent application behaviour, which is highly driven by the reuse of core features rather than building it from scratch.

VI. CONCLUSION AND FUTURE SCOPE

Built with Kotlin Multiplatform in mind, this application aims to bring ease and user-friendliness to anyone navigating a university campus, even without a GPS or internet connection. It navigates the user around a series of locally stored checkpoints, providing visual and textual aids to offer the most seamless way to navigate the campus. With Kotlin Multiplatform, it is easy to reuse the code, saving a significant amount of development effort while allowing for native experiences with both Android and iOS. Notwithstanding this fact, the application does not have real-time location tracking capabilities; however, it still saves its users time by providing a clear indication and facilitating the easy recognition of landmarks.

Even if the application serves its general purpose, there are still areas for development, particularly in terms of adding not only functionality but also user comfort.

Arguably, now it is the most promising area of development in integrating augmented reality. In other words, augmented reality would be implemented by overlaying virtual directions and markers over a device's view of the real world through a camera. That would make visual navigation even more immersive and dynamic than it currently is. Users would not have to match the Checkpoint images against their surroundings because they will be constantly provided with visual cues through AR, which will help them navigate across campus. Indoor navigation might be another future area of interest that could attract interest in the future. For now, this application is practical outdoors, where checkpoints can be easily seen. On the other hand, it would be challenging for large buildings or indoor spaces, as GPS signals are often weak or unavailable indoors. Future versions would be developed with the implementation of indoor navigation techniques, such as Bluetooth beacons, Wi-Fi positioning, or image recognition, to assist users in navigating indoors. This would enhance the practical use of the application, particularly in complex environments such as libraries, lecture halls, or multilevel buildings.

Retrieval Number: 100.1/ijitee.H111614080725
DOI: 10.35940/ijitee.H1116.14090825
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Additionally, one could go even further with more personalised options for the user. Since campuses can vary in size and topography, users can be offered options such as the shortest distance, accessibility, and avoidance of highly populated areas, among others. Building feedback mechanisms alongside such recommendations would help facilitate overall improvement, providing better route guidance or more accurate instructions by learning user behaviour over time. It will, therefore, also be possible to integrate it with real-time data systems, while maintaining an offline mode for enhanced functionality. A good example is the inclusion of event-based notifications that alert students whenever routes are available or an event is happening on campus, thereby making the application more dynamic and responsive.

Also, integration with other campus resources, like timetables or event schedules, would make it possible to be able to offer suggestions at appropriate and relevant times, making the app more useful in activity-peaked occasions, such as the

Orientation week or exam season. Finally, while arguably at the very core of user guidance for offline large-campus scenarios, the scope that can be reached with the Campus Navigation App indeed seems extremely broad. As augmented reality, wide-scale indoor navigation, and real-time data integration are added to the list, the application will become more forceful and multi-functional in covering a wider range of navigation needs within academic institutions.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/Competing Interests:** Based on my understanding, this article does not have any conflicts of interest.
- **Funding Support:** This article has not been sponsored or funded by any organization or agency. The independence of this research is a crucial factor in affirming its impartiality, as it was conducted without any external influence.
- **Ethical Approval and Consent to Participate:** The data provided in this article is exempt from the requirement for ethical approval or participant consent.
- **Data Access Statement and Material Availability:** The data and materials supporting the findings of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Gryzun, L.E., Shcherbakov, O.V. and Bida, B.O., 2023. Development of an information system for navigation on modern university campuses. CTE Workshop Proceedings [Online], 10, pp.380–398. DOI: https://doi.org/10.55056/cte.566
2. R. A. Abbas Helmi, H. A-P Ravichandran, A. Jamal and M. N. Mohammed, "Design and Development of Indoor Campus Navigation Application," 2022 IEEE 10th Conference on Systems, Process & Control (ICSPC), Malacca, Malaysia, 2022, pp. 77-82. DOI: https://doi.org/10.1109/ICSPC55597.2022.10001791
3. Ang, K. J. C., Carag, S. L., Jacinto, D. E. M., Lunasco, J. O. R., & Cabatuan, M. K. (2018). Design and Development of Google Glass-Based Campus Navigation System. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-5), 75–81. URL: https://jtec.utem.edu.my/jtec/article/view/3635
4. Sparsh, Abhishek Kumar Jadaun, Arjun Pandit, Jabir Ali, Mohd.Shariq, "Offline Navigation System for Campus", International Journal of Advances in Engineering and Management (IJAEM) Volume 5, Issue 1 Jan. 2023, URL:https://ijaem.net/issue_dcp/Offline%20Navigation%20System%20for%20Campus.pdf
5. Anpat, V.; Shewale, A.; Bhangale, Y. Campus Navigation on Android Platform. *Int. J. Sci. Technol. Eng.* 2016, *2*, 452–458. URL: https://ijsart.com/campus-navigation-on-android-platform-457
6. K. Subaramaniam, C. Y. Lee and C. T. Ong, "UCSI Campus Navigator," 2019 IEEE 9th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia, 2019, pp. 343-348, DOI: https://doi.org/10.1109/ICSEngT.2019.8906424
7. U. T. Chitra, R. K, H. M and P. S, "Indoor Campus Navigation using Web Application System for Seamless University Mobility," 2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS), Chennai, India, 2023, pp. 1-4, DOI: https://doi.org/10.1109/ICCEBS58601.2023.10448673
8. Li, J., & Yin, Z., & Timorabadi, H. S. (2024, June), Board 44: CampNav: A System for Inside Buildings and Campus Navigation. Paper presented at the 2024 ASEE Annual Conference & Exposition, Portland, Oregon. DOI: 10.18260/1-2—47031 URL: https://peer.asee.org/47031
9. K. Manning and M. Hosein, "An Android-Based Navigation Application for A Tertiary Campus," 2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 2021, pp. 1-5, DOI: https://doi.org/10.1109/ICMNWC52512.2021.9688402
10. C. Chotbenjamaporn, A. Chutisilp, P. Threethanuchai, S. Poolkrajang, M. Tuwawit, P. Laowong, A. Tirajitto, E. Wang, R. Muangsiri, A. Compeecharoenporn et al., "A web-based navigation system for a smart campus with air quality monitoring", *2019 IEEE International Smart Cities Conference (ISC2)*, pp. 581-586, 2019. DOI: https://doi.org/10.1109/ISC246665.2019.9071669
11. Nagy, R., 2022. Simplifying Application Development with Kotlin Multiplatform Mobile: Write robust native applications for iOS and Android efficiently. Packt Publishing Ltd.
12. Singh M. and Shobha G. 2021. Comparative Analysis of Hybrid Mobile App Development Frameworks. International Journal of Soft Computing and Engineering (IJSCE). 10, 6 (July 2021). DOI: https://doi.org/10.35940/ijsce.F3518.0710621
13. Shah K., Sinha H. and Mishra P. 2019. Analysis of Cross-Platform Mobile App Development Tools. 5th International Conference for Convergence in Technology (12CT) (Pune, India, March 29-31, 2019). IEEE'19. IEEE, New Jersey, 1-2. DOI: https://doi.org/10.1109/I2CT45611.2019.9033872
14. Kotlin Multiplatform Documentation and libraries-: https://kotlinlang.org/docs/multiplatform.html#desktop-applications

AUTHOR'S PROFILE

Prathamesh Bhusekar is a final-year student pursuing a Bachelor of Engineering in Information Technology from Vidyalankar Institute of Technology, Mumbai. His research interests include Application Development and Cloud Computing.

Shivanand Manyar is a final-year student pursuing a Bachelor of Engineering in Information Technology from Vidyalankar Institute of Technology, Mumbai. His research interests include Machine Learning, Artificial Intelligence, and IoT.

Hrishita Thanekar is a final-year student pursuing a Bachelor of Engineering in Electronics and Telecommunications Engineering from Vidyalankar Institute of Technology, Mumbai. Her research interests include Machine learning, Artificial Intelligence, and IoT.

Dr. Vidya Chitre is a highly experienced Professor in Information Technology with over 27 years of expertise at Vidyalankar Institute of Technology in Wadala, Mumbai. Dr. Chitre holds a Ph.D. in Information Technology from the University of Mumbai and has specialised knowledge in machine learning. Her

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

research and teaching interests encompass algorithms, data mining, artificial intelligence, and predictive modelling. With a passion for nurturing students' skills, Dr. Chitre has guided 2 PhD and 18 Master's students. She has published 47 papers in reputed journals and conferences, made significant contributions to the field, and inspired many aspiring IT professionals.