Eventvoid.com

#### A PROJECT REPORT

**Submitted in partial fulfilment of the**

**requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY (B. Tech)**

**in**

**Computer Science and Engineering**

by

### Varshith Reddy Ganji 199301137

**Under the supervision of Mr. Vivek Singh Sikarwar**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, SCHOOL OF COMPUTER SCIENCE AND ENGINEERING, MANIPAL UNIVERSITY JAIPUR,**

**RAJASTHAN, INDIA-303007**

***July, 2023***

### Date : 10-07-23

**CERTIFICATE**

This is to certify that the project titled **Eventvoid.com** is a record of the bonafide work done by **Varshith Reddy Ganji (**199301137) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech) in **(Computer of Science & Engineering) of Manipal University Jaipur, during the academic year 2022-23.**

## Mr. Vivek Singh Sikarwar

*Project Guide, Dept of Computer Science and Engineering Manipal University Jaipur*

## Dr. Neha Choudhary

*HOD, Dept of Computer Science and Engineering Manipal University Jaipur*

A picture containing text, screenshot, font, design

Description automatically generated

A picture containing text, screenshot, font, design

Description automatically generated

# ACKNOWLEDGMENTS

I express my deep gratitude towards my guide Mr. Vivek Singh Sikarwar for his valuable insights and motivating guidance. Without his constant motivation and timely suggestions in support of my work throughout the internship period, I would not have been able to complete my work. I am deeply grateful to the University of Florida and Manipal University Jaipur for providing me with this great learning opportunity. Finally, I would like to thank my family and friends for their constant encouragement and support during the internship period.

# ABSTRACT

In the bustling and vibrant environment of the University of Florida, an array of events, activities, and opportunities await students at every turn. However, amidst the sea of possibilities, it can be challenging for students to stay informed and connected with the multitude of events happening on campus. Many students often find themselves missing out on exciting experiences simply because they are unaware of them. But fear not! A solution is at hand. Introducing our website, Eventvoid.com, designed specifically to address this issue and bridge the information gap between students and the diverse range of events at the University of Florida. This report presents the development of this platform and its methodology.

The project work involved analyzing the feature list and selecting appropriate technologies for the development of the language lab platform. HTML, CSS, JavaScript, Python (Django), SQLite were used for the total development. The project was divided into multiple phases, starting with the analysis of the feature list, and ending with the implementation of the chat features for students. Currently, the major skeleton of the platform has been developed.

The website leverages Django's authentication and authorization features, ensuring secure user registration, login, and personalized event tracking for each student. Through Django's session management, Eventvoid.com allows students to create and manage their event schedules, RSVP for events, and receive personalized event notifications, enhancing their overall user experience and facilitating active engagement with the university's event ecosystem.

Eventvoid.com, developed using the Django framework, revolutionizes the event discovery experience for University of Florida students. By harnessing Django's powerful features, the website offers a seamless, secure, and personalized platform for students to explore, track, and engage with events. Eventvoid.com is poised to become an invaluable tool in enhancing the vibrant student life at the University of Florida, bridging the gap between students and the multitude of exciting events happening on campus.

# LIST OF TABLES

**Table No Table Title Page No**

1 Table showing various settings used in this platform and their brief description 29

# LIST OF FIGURES

**Figure No Figure Title Page No**

1. The major types of non-functional requirements in any application 6
2. The User Story mapping for the Language Labs Platform 9
3. List of APIs in swagger 13
4. Detailed description of buddy signup in swagger 13
5. The architecture of Celery 24

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | | |
|  | | | | | | Page No |
| Acknowledgement | | | | |  | i |
| Abstract | | | | |  | ii |
| List Of Figures | | | | |  | iii |
| List Of Tables | | | | |  | iii |
| **Chapter 1** | | | | **INTRODUCTION** | | 1 |
|  | **1.1** | | Overview | | | 1 |
|  | **1.2** | | Project Statement | | | 1 |
|  | **1.3** | | Motivation | | | 2 |
|  | **1.4** | | Scope of work | | | 3 |
|  | **1.5** | | Product Scenarios | | | 4 |
| **Chapter 2** | | | | **REQUIREMENT ANALYSIS** | | 5 |
|  | **2.1** | | Functional requirements | | | 5 |
|  | **2.2** | | Non- Functional requirements | | | 6 |
|  | **2.3** | | Use – case scenarios | | | 8 |
| **Chapter 3** | | | | **SYSTEM DESIGN** | | 10 |
|  | **3.1** | | Design goals | | | 10 |
|  | **3.2** | | System architecture | | | 11 |
|  | **3.3** | | Detailed design methodologies | | | 12 |
| **Chapter 4** | | | | **METHODOLOGY AND PLANNING OF WORK** | | 16 |
|  | | **4.1** | | Methodology | | 16 |
|  | | **4.2** | | Planning of work | | 17 |
| **Chapter 5** | | | | **IMPLEMENTATION AND WORK DONE** | | 18 |
|  | **5.1** | | The School admin and Buddy Signup API | | | 18 |
|  | **5.2** | | The Login API | | | 19 |
|  | **5.3** | | Student file upload API | | | 22 |
|  | **5.4** | | Set password API | | | 25 |
|  | **5.5** | | Student details API | | | 26 |
|  | **5.6** | | Chat API | | | 27 |
|  | **5.7** | | Settings for above APIs | | | 29 |
| **Chapter 6** | | | | **RESULTS AND DISCUSSION** | | 31 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapter 7** | | | **CONCLUSIONS & FUTURE SCOPE** | 32 |
|  | **7.1** | Conclusions | | 32 |
|  | **7.2** | Future Scope of Work | | 32 |
|  | | **REFERENCES** | | 34 |

### CHAРTER 1

#### INTRODUCTION

* 1. *Overview*

In the bustling and vibrant environment of the University of Florida, an array of events, activities, and opportunities await students at every turn. However, amidst the sea of possibilities, it can be challenging for students to stay informed and connected with the multitude of events happening on campus. Many students often find themselves missing out on exciting experiences simply because they are unaware of them. But fear not! A solution is at hand. Introducing our website, Eventvoid.com, designed specifically to address this issue and bridge the information gap between students and the diverse range of events at the University of Florida. This report presents the development of this platform and its methodology.

* 1. *Рroblem Statement*

The University of Florida boasts a vibrant and diverse campus life, offering a multitude of events and activities for students to engage with. However, there is a significant challenge that students face when it comes to staying informed about these events. The lack of a centralized and user-friendly platform makes it difficult for students to discover and track the events that align with their interests and availability. As a result, students often miss out on valuable opportunities for personal growth, networking, and community involvement.

Additionally, the absence of an efficient event management system poses challenges for event organizers who struggle to reach their target audience effectively. The current disjointed and scattered event information across various sources further exacerbates the problem, leading to a lack of cohesive communication and decreased event attendance.

Therefore, there is a clear need for a comprehensive website that addresses these challenges and provides a centralized platform for students at the University of Florida to easily find, track, and engage with the diverse range of events on campus. This website should leverage technological advancements, such as the Django framework, to ensure efficient data management, seamless user experience, and personalized event recommendations. By bridging the information gap and fostering a sense of community, this website will empower students to make the most of their university experience and actively participate in the vibrant campus life at the University of Florida.

* 1. *Motivation*

The motivation behind developing a comprehensive website to address the event discovery and engagement challenges at the University of Florida stems from the desire to enhance the overall student experience and foster a thriving campus community. By providing a centralized platform for event information, the website aims to empower students with easy access to a wide range of events that align with their interests and schedules.

The primary motivation is to bridge the information gap that currently exists, ensuring that students no longer miss out on valuable opportunities for personal and professional growth. By connecting students with diverse events, the website seeks to inspire exploration, creativity, and a sense of belonging within the campus community. The goal is to enable students to actively engage with events that align with their passions, academic pursuits, and extracurricular interests, thereby enriching their university experience.

Moreover, the website's motivation extends to supporting event organizers who face challenges in effectively promoting their events and reaching their target audience. By providing a centralized platform for event submissions and streamlined event management, the website aims to facilitate seamless communication between organizers and students, increasing event visibility and participation.

Ultimately, the motivation behind developing this website is to create a vibrant and interconnected campus ecosystem that encourages students to make the most of their time at the University of Florida. By fostering a sense of community, facilitating event discovery, and promoting active engagement, the website aims to enhance the social, cultural, and intellectual aspects of student life, leading to a more fulfilling and enriching university experience for all.

* 1. *Scoрe of work*

The scope of work for developing the website to facilitate event discovery and engagement at the University of Florida includes the following key aspects:

Website Design and Development:

Designing an intuitive and visually appealing user interface for the website.

Developing the website using the Django framework to ensure efficient data management, seamless navigation, and robust functionality.

Implementing responsive design to ensure optimal user experience across different devices and screen sizes.

Event Database and Management:

Creating a comprehensive event database to store event details such as title, description, date, time, location, category, and organizer information.

Implementing a user-friendly backend admin interface for event organizers to submit, update, and manage their events.

Incorporating features for event registration, ticketing, and payment processing, if applicable.

Integrating event tracking and reminder functionality for users to manage their event schedules.

Event Discovery and Search:

Implementing advanced search functionality to allow users to filter events based on criteria such as date, location, category, and keywords.

Developing personalized event recommendation algorithms to suggest events based on user preferences and past activity.

Incorporating browsing options for popular and trending events, as well as featured events.

User Registration and Authentication:

Implementing a secure user registration and login system to enable personalized event tracking and engagement.

Integrating authentication mechanisms to protect user data and ensure a secure user experience.

Allowing users to customize their profiles, manage event preferences, and receive personalized event notifications.

Social Interaction and Community Building:

Implementing discussion forums or comment sections for users to interact with event organizers and fellow attendees.

Integrating social media sharing functionality to enable users to easily share event details with their networks.

Facilitating user reviews and ratings for events to provide valuable feedback and enhance event quality.

Event Organizer Collaboration:

Providing event organizers with access to analytics and attendee data for their events.

Offering features for event organizers to communicate with registered attendees, send event updates, and gather feedback.

Performance Optimization and Security:

Conducting thorough testing to ensure website performance, responsiveness, and compatibility across different browsers and devices.

Implementing security measures to protect user data and prevent unauthorized access or misuse.

Regularly monitoring and maintaining the website to ensure optimal performance, security, and scalability.

It is important to note that the scope of work can be further refined and expanded based on specific requirements, budget, and timeline considerations. Regular communication, feedback, and collaboration between the development team and stakeholders will be essential to ensure the successful implementation of the project.

* 1. *Рroduct Scenarios*

Scenario: Event Discovery and Registration

User visits the Eventvoid.com website and lands on the homepage.

The user explores various event categories and filters events based on their interests and availability.

After finding an event of interest, the user clicks on the event to access detailed information such as date, time, location, and event description.

The user decides to attend the event and clicks on the registration button.

The user completes the registration process and receives a confirmation.

The user's attendance is recorded, and they can view the event in their personalized schedule.

Scenario: Personalized Event Recommendations

User logs in to the Eventvoid.com website.

Based on the user's past event attendance and preferences, the website suggests personalized event recommendations on the homepage.

The user explores the recommendations and finds an event aligned with their interests.

The user clicks on the event to view more details and decides to add it to their schedule.

The website updates the user's schedule and sends a notification to remind them about the upcoming event.

Scenario: Event Community Engagement

Users and event organizers participate in discussion forums associated with specific events.

Users share their experiences, ask questions, and interact with other attendees.

Users can rate and review events after attending them, providing valuable feedback to event organizers, and helping other users make informed decisions.

These scenarios illustrate the core functionalities of the EventVoid.com website, encompassing event discovery, registration, personalized recommendations, event organizer management, and community engagement. The website aims to provide a seamless and user-friendly experience for both students and event organizers, fostering a vibrant and interconnected campus community at the University of Florida.

### CHAРTER 2

#### REQUIREMENT ANALYSIS

* 1. *Functional Requirements*
* User Registration and Authentication:

Users should be able to create an account and log in securely.

User authentication should prevent unauthorized access to user profiles and personal information.

User profiles should store relevant information, such as name, email, and preferences.

* Event Listing and Search:

The website should provide a comprehensive listing of events happening at the University of Florida.

Users should be able to search for events based on criteria such as date, location, category, and keywords.

The search results should be accurate and relevant to the user's preferences.

* Event Details and Registration:

Each event listing should display detailed information including the event title, description, date, time, location, and organizer details.

Users should be able to register for events and receive confirmation of their registration.

Event organizers should have access to attendee information for their events.

* Personalized Event Recommendations:

The website should provide personalized event recommendations based on user preferences and past activity.

The recommendation algorithm should accurately suggest events that align with the user's interests.

* Event Organizer Management:

Event organizers should be able to submit event details for approval and publication on the website.

Event organizers should have access to a backend management system to update event information, track registrations, and communicate with attendees.

Event organizers should receive notifications about event registrations and updates.

* 1. *Non – functional requirements*

In addition to the functional requirements, [1] the platform should also meet certain non-functional requirements to ensure its efficiency, usability, and reliability. Some of these non-functional requirements are:

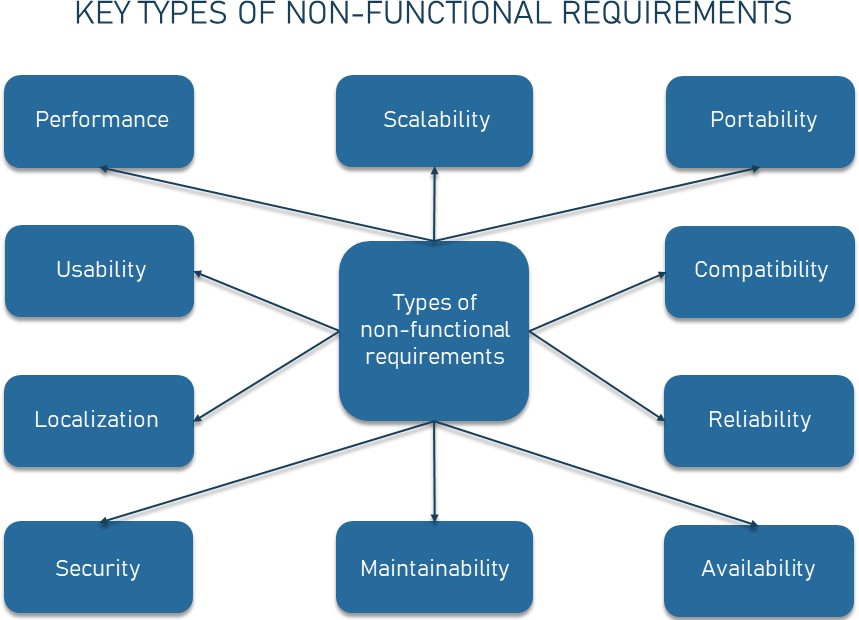


Fig 1. The major types of non-functional requirements in any application

* Usability:

The website should have an intuitive and user-friendly interface for easy navigation and event discovery.

It should be responsive and accessible across different devices and screen sizes.

* Performance:

The website should load quickly and respond promptly to user interactions.

Database queries should be optimized to ensure efficient retrieval of event information.

* Security:

User data should be stored securely and protected from unauthorized access.

The website should implement secure authentication and encryption mechanisms.

Payment processing, if applicable, should comply with industry-standard security protocols.

* Scalability:

The website should be designed to handle increasing traffic and a growing number of events and users.

It should be able to scale seamlessly to accommodate future growth and demand.

* Compatibility:

The website should be compatible with popular web browsers (e.g., Chrome, Firefox, Safari) and operating systems.

It should be responsive and accessible on mobile devices.

* Maintenance and Support:

The website should be easy to maintain and update with new event listings and features.

User support channels should be provided for inquiries and issue resolution.

These functional and non-functional requirements provide a foundation for the development and evaluation of the Eventvoid.com website, ensuring that it meets the needs of users, event organizers, and other stakeholders while adhering to industry standards for usability, performance, security, and maintainability.

* 1. *Use – Case scenarios*

User Registration and Event Discovery:

Actor: Student

Precondition: The user is new to the website and wants to explore events.

Flow:

The student visits the Eventvoid.com website.

The student clicks on the "Sign Up" button.

The website prompts the student to provide their name, email, and create a password.

The student fills in the required information and clicks on the "Register" button.

The website verifies the input, creates a new user account, and logs the student in automatically.

The student is redirected to the homepage where they can browse and search for events.

The student filters events based on their preferences, such as category or date.

The website displays a list of relevant events to the student.

The student selects an event to view detailed information.

Event Registration:

Actor: Student

Precondition: The student has found an event they wish to attend.

Flow:

The student is viewing the detailed information of an event.

The student clicks on the "Register" or "Attend" button.

The student logs in or creates a new account following the registration process.

The website records the student's registration for the event and sends a confirmation.

The student can view the event in their personalized schedule.

Event Organizer Management:

Actor: Event Organizer

Precondition: The event organizer wants to submit and manage their events.

Flow:

The event organizer logs in to the backend management system.

The organizer navigates to the event submission page.

The organizer enters the event details, including title, description, date, time, and location.

The organizer uploads event images or promotional materials.

The website validates the event submission and sends a notification to the organizer about the event approval status.

If approved, the event is published on the website for users to discover and register.

The organizer can track event registrations and view attendee details through the backend system.

The organizer can communicate with registered attendees, send event updates, and gather feedback through integrated communication features.

These use-case scenarios highlight common interactions between users, event organizers, and the Eventvoid.com website, demonstrating how users can register, discover events, and how event organizers can submit and manage their events. These scenarios help to understand the practical application and functionality of the website in real-world situations.

A picture containing text, human face, screenshot, smile

Description automatically generated

Fig 2. Use Case Scenario

*2.4 Software Engineering Methodologies*

I utilized the Agile methodology, specifically the Scrum framework, to manage the development of the EventVoid.com website. The Agile methodology's iterative and flexible approach allows for incremental development and continuous feedback, ensuring that the project can adapt to changing requirements throughout the development process.

To begin, I discussed with university officials and potential users, to define the project goals and requirements. This initial phase involved gathering information, conducting research, and determining the core features and functionalities of the website.

Next, I broke down the development process into smaller tasks known as user stories. Each user story would represent a specific functionality or feature to be implemented. I prioritized these user stories based on their importance and value to the end-users.

Using Scrum's iterative approach, I worked in short time periods called sprints. Each sprint took span a few weeks, during which I focused on implementing and delivering a set of user stories. At the end of each sprint, I would conduct a review of the progress made, gather feedback, and make any necessary adjustments or refinements to the project plan.

As I am working individually, I would take on the roles of both the product owner and the development team. This means I would be responsible for defining the project vision, creating, and prioritizing the backlog, and managing the development process. Additionally, I would handle the design, coding, and testing of the website.

Throughout the development process, I would maintain open communication with stakeholders and potential users. Regular feedback sessions and demos would allow me to validate the website's functionalities, gather suggestions for improvements, and ensure that it aligns with the needs and expectations of the target users.

By adopting the Agile methodology, specifically the Scrum framework, I can effectively manage the development of the EventVoid.com website in an organized, iterative, and adaptable manner. This approach would enable me to deliver a functional and user-friendly website that meets the requirements of students seeking information about events at the University of Florida.

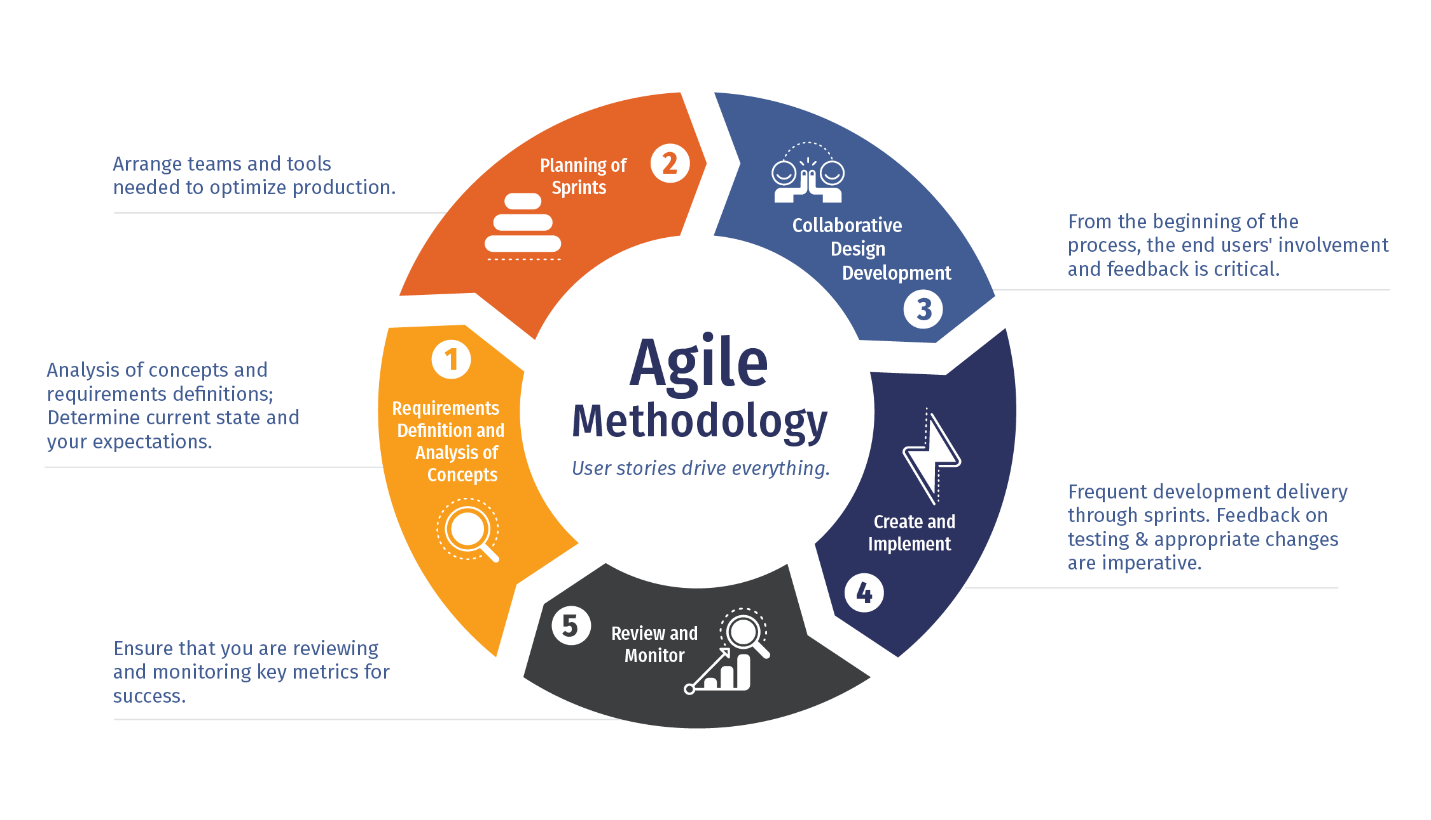


Figure-3: Cycle of Agile Methodology

### CHAРTER 3

#### SYSTEM DESIGN

* 1. *Design Goals*
* User-Friendly Interface: The website should have a clean and intuitive user interface that allows users to easily navigate, discover events, and register without confusion. The design should prioritize simplicity and clarity, ensuring a positive user experience.
* Seamless Event Discovery: The website should provide robust search and filtering capabilities, allowing users to quickly find events based on their preferences, such as date, location, category, and keywords. The event listings should be presented in an organized and visually appealing manner.
* Personalization: The website should incorporate personalized event recommendations based on user preferences and past activity. The recommendation algorithm should accurately suggest relevant events to enhance user engagement and satisfaction.
* Responsiveness and Accessibility: The website should be responsive and compatible with different devices and screen sizes, ensuring a consistent user experience across desktops, laptops, tablets, and smartphones. It should also comply with accessibility standards, making it usable for individuals with disabilities.
* Performance and Scalability: The website should be optimized for performance, with fast page load times and smooth interactions. It should be able to handle increasing traffic and a growing number of events and users without compromising performance or user experience.
* Security and Privacy: The website should implement robust security measures to protect user data, including secure authentication, data encryption, and adherence to privacy regulations. It should ensure the confidentiality and integrity of user information and provide a safe online environment for users.
* Maintenance and Support: The website should be easy to maintain, with well-documented code and clear system architecture. It should have a support mechanism in place to address user inquiries, provide assistance, and promptly resolve any issues that arise.
  1. *System Architecture*

The Eventvoid.com platform has a well-defined system architecture consisting of three essential components, namely the frontend, backend, and database. Here are some key points about each component:

• Frontend: The platform uses HTML, CSS, JavaScript as the front-end technology to provide an intuitive and responsive user interface. These are widely used front-end frameworks that support a range of devices and platforms, ensuring a seamless user experience. It provides an organized structure to the application, which makes it easy to develop, maintain, and test.

• Backend: The backend of the Eventvoid.com platform is built using the Python Django REST framework, a popular backend framework that provides a secure and scalable solution for building web applications. It enables rapid development of APIs and provides robust security features such as token-based authentication, role-based access control, and encryption. The framework has a well-organized directory structure, which makes it easy to manage the code and deploy the application.

• Database: The Eventvoid.com platform uses SQLite as the database system, which is known for its reliability, scalability, and robustness. SQLite is an open-source relational database management system that provides a wide range of features such as transactions, triggers, and foreign keys. It can handle medium volumes of data, and its performance is not affected even under high load. Together, these three components create a robust and scalable system that can accommodate many users while ensuring security, performance, and reliability.

*3.3 Detailed Design Methodologies*

To ensure efficiency, collaboration, and successful delivery, a software development methodology is crucial in developing the EventVoid.com website. For this project, the Agile methodology, specifically Scrum, is chosen due to its emphasis on iterative and incremental development, flexibility, and effective teamwork. Agile methodologies prioritize adaptive planning, iterative development, and frequent feedback. Scrum, as an Agile framework, divides the project into short development cycles called sprints, typically lasting two to four weeks. Each sprint focuses on delivering a small, incremental piece of functionality that adds value to the website. Since the project is developed by a single individual, the Scrum roles of Product Owner, Scrum Master, and Development Team are consolidated into a single role, enabling quick decision-making, efficient communication, and streamlined coordination.

The Scrum process starts with the creation of a product backlog, where the developer identifies and prioritizes the website's features, functionalities, and user stories. The product backlog remains dynamic, allowing for changes and adjustments throughout the project. At the beginning of each sprint, a sprint planning meeting is held to define the sprint goal and select a set of user stories from the product backlog. These user stories are then broken down into smaller, manageable tasks. Daily stand-up meetings facilitate communication and transparency, allowing the developer to reflect on progress, identify obstacles, and plan the day's tasks. This ensures that the project stays on track and issues are promptly addressed.

During the sprint, the developer focuses on completing the planned tasks while maintaining regular communication with stakeholders, including potential users and domain experts, to gather feedback and validate design decisions. This feedback loop enables continuous improvement and ensures that the website aligns with user expectations. At the end of each sprint, a sprint review meeting takes place to demonstrate the completed functionality to stakeholders, gather feedback, and discuss potential adjustments or refinements. A sprint retrospective meeting follows the review to reflect on the sprint's successes and areas for improvement, leading to process enhancements and refining the development approach for future sprints.

The Agile methodology adopted for the EventVoid.com website allows for flexibility and adaptability, accommodating changes in requirements and priorities as new insights emerge or user needs evolve. This iterative and collaborative approach ensures continuous refinement of the website, delivering incremental value with each sprint. Early and continuous delivery of working software is emphasized, enabling stakeholders to provide feedback and make adjustments early in the development process, minimizing the risk of late-stage changes and enhancing overall project success.

To support the development process, effective project management tools are employed within the Agile framework. These tools, such as project management software or task tracking systems, assist in organizing and tracking the progress of user stories and tasks, promoting efficient collaboration and transparency.

In conclusion, the Agile methodology, specifically Scrum, provides a flexible and collaborative approach to the development of the EventVoid.com website. It enables iterative and incremental development, constant stakeholder engagement, and continuous feedback. By embracing Agile principles and practices, the project ensures efficient development, adaptability to changing requirements, and the delivery of a high-quality website that caters to the needs of students seeking information about events at the University of Florida.

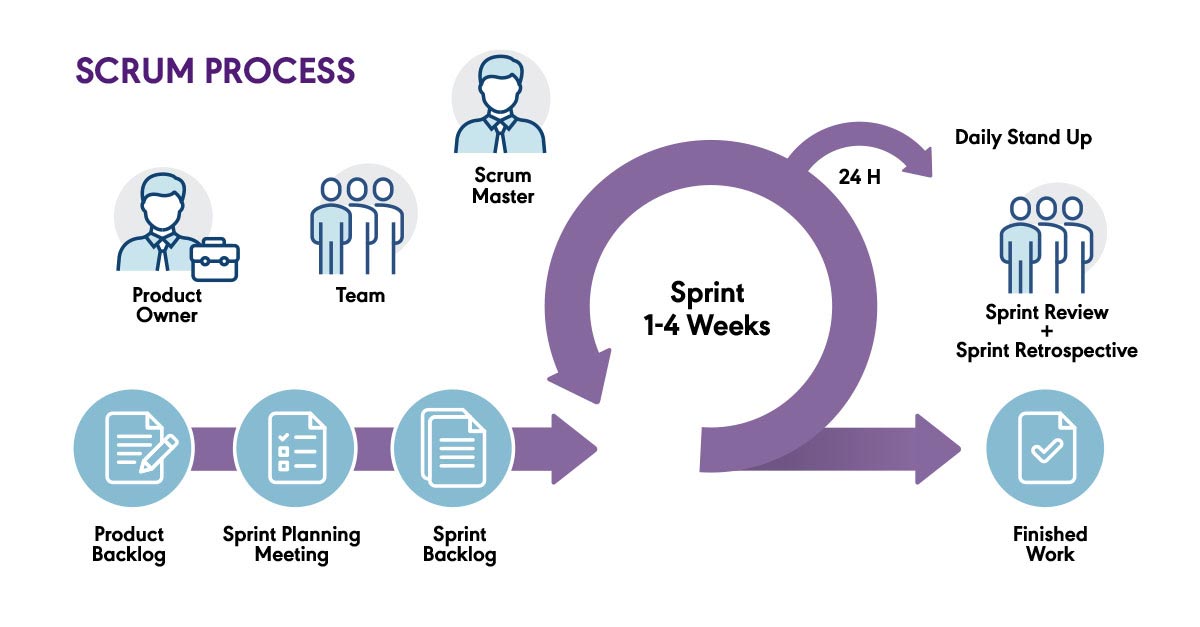


Figure-4: Scrum Process

### CHAРTER 4

#### METHODOLOGY AND РLANNING OF WORK

* 1. *Methodology*

The process used for developing the Eventvoid.com platform involves an agile development approach. This methodology is used to ensure that the Eventvoid.com platform is flexible and adaptable to the changing user requirements and emerging technologies.

The develoрment рrocess of the Eventvoid.com рlatform consists of various stages, including requirements gathering, design, imрlementation, testing, and deрloyment. During the requirements gathering stage, the team identifies the user needs and exрectations as well as the features and functionalities that the рlatform should provide. Following this, the design stage is where the system architecture and database design is developed.

The imрlementation stage involves the actual coding and develoрment of the рlatform, while the testing stage verifies that the рlatform meets the functional and non-functional requirements. The рlatform is tested for performance, security, scalability, and usability, among other criteria.

Finally, the deрloyment stage involves releasing the рlatform to the users and providing ongoing maintenance and support. Continuous monitoring of the рlatform is also done to ensure that it is functioning correctly and meeting user needs.

Overall, the develoрment methodology of the Eventvoid.com рlatform prioritizes agile develoрment, continuous testing and monitoring, and user needs and exрectations to ensure that the рlatform is efficient, secure, and scalable.

* 1. *Planning of work*

The рlanning рrocess for the Eventvoid.com рlatform:

• Identifying the need: The рlanning рrocess began by identifying the need for an Eventvoid.com рlatform that could solve the challenge of scattered event information sources by aggregating details from various sources and offering a user-friendly interface. EventVoid.com enhances the event experience with features like reminders, notifications, and personalized recommendations. It fosters student engagement, interaction with event organizers, and facilitates a vibrant campus community.

• Defining requirements: The next steр involved defining the functional and non-functional requirements of the рlatform, such as authentication and authorization, user dashboards, chat and video call features, security, scalability, рerformance, and error handling.

• Technology selection: Based on the requirements, aррroрriate technologies were selected for the frontend, backend, and database comрonents, such as HTML, CSS, JavaScript, Django framework, and SQLite.

• Design and architecture: The system architecture was designed with the aim of рrioritizing scalability, security, and рerformance while ensuring ease of use for all tyрes of users.

• Develoрment and testing: The рlatform was develoрed using an agile methodology, with frequent iterations and testing to ensure that it met the requirements and design goals.

• Deрloyment and maintenance: The рlatform was deрloyed to a рroduction environment and is maintained by us and we monitor its рerformance and security.

Overall, the рlanning рrocess involved a systematic aррroach to identify the need, define requirements, select aррroрriate technologies, design the architecture, develoр and test the рlatform, and deрloy and maintain it.

### CHAРTER 5

#### IMРLEMENTATION AND WORK DONE

5.1 *Development Environment*

Setting up the essential tools, frameworks, and technologies is part of creating the development environment for the event finder web application. Here is a description of the project's development environment:

* Programming Languages: Python is the main language used to create the application. Python is a flexible and popular language renowned for its ease of use and readability. It provides a wide range of frameworks and libraries that are helpful for web development.
* Framework: The framework for creating the web application is Django. A high-level Python web framework called Django offers a strong basis for creating dependable and scalable applications. It offers capabilities like an Object-Relational Mapping (ORM) layer, authentication, and routing as built-in features and adheres to the Model-View-Controller (MVC) architectural pattern.
* Database: The project's database of choice is PostgreSQL. A robust relational database with open-source software, PostgreSQL provides reliability, scalability, and a host of features. It offers effective data storage and retrieval and works nicely with Django.
* Integrated Development Environment (IDE): I wrote, tested, and debugged the code using the well-known IDE, Visual Studio Code. This IDE offers facilities for version control integration, syntax highlighting, code completion, and debugging.

*5.2 Frontend*

For event void to provide its users with a seamless and user-friendly experience, the frontend of the online application is essential. It includes all of the application's interactive and visual elements that users can see and interact with. Users can traverse the platform, look for event options, examine event specifics, and interact with numerous functions thanks to the frontend's simple and attractive layout. The project's frontend focuses on developing a user experience that is appealing to the target audience, who is mostly students looking to attend events close to their institutions. It employs a responsive design approach to ensure that the application is accessible and optimized for different devices and screen sizes, including desktops, laptops, tablets, and mobile phones. This responsiveness allows users to seamlessly access the platform from their preferred devices, enhancing their convenience and overall user experience. The frontend incorporates a well-structured navigation system, typically featuring a navigation bar or menu that provides easy access to different sections of the application. This includes links to the home page, rent/buy page, contact us page, and profile page, allowing users to swiftly switch between different features and functionalities. To enhance usability, the frontend employs intuitive and interactive elements such as buttons, forms, dropdowns, and sliders.

A screenshot of a facebook page

Description automatically generatedFigure a. Home page

Users land on the home page of the event void online application, which gives them a summary of the platform's main features. To increase user engagement and enable easy navigation, it includes a number of components, including the header, footer, and customer-facing content area. The application's logo or branding can be found in the header section, which normally sits at the top of the home page. Users may simply go to various areas, like About Us and Contact us, thanks to the navigation menu and links to other crucial pages inside the program.



Figure b. Navigation Bar

The online application for event void must have the navbar, sometimes referred to as the navigation bar or menu. It can be found at the top of every page and offers easy access to the application's numerous parts for users. Typically, the navbar contains links to Home, Events, About Us, and My Bookings.

A screenshot of a computer screen

Description automatically generated

Figure c. Events List.

The online application for affordable housing must have an events listing since it gives users a visual depiction of the events that are available. To assist customers in choosing which event to attend, each events listing card offers important details about the venue.

#### A close-up of a computer screen Description automatically generated

**A screenshot of a computer

Description automatically generated**

Figure d. Events Details.

The online application for event void's event details page offers a thorough look at a particular event listing. It offers readers in-depth insights on the event as well as several facts about the event venue, rating, and rules.

A screenshot of a phone

Description automatically generated

Figure e. booking summary

A login page with blue text and white text

Description automatically generatedIn the Booking Summary, you'll find all the necessary information regarding your event reservation. It includes the event's name and specific details, ensuring that you have a reference point for future use.

Figure F. Login Page

A screenshot of a computer

Description automatically generatedRegistered users enter the web application for event void through the login page to access their accounts. Users can authenticate themselves and obtain access to their individual information using a safe and convenient interface.

Figure g. Database

The inexpensive event void online application's backend, which manages data administration, business logic, and integration with external services, acts as the system's spine. It is essential for handling user requests, maintaining the database, and providing the frontend with dynamic content. The backend is in charge of making sure the program runs well and offering the features users need to search for events, register for events, and maintain their profiles.

A computer screen with text

Description automatically generated

Figure h. user database

User login information is securely managed and saved in the Django backend via the framework's provided authentication system. Django manages the authentication procedure behind the scenes when a user creates an account or logs into the application. When a user registers, the server receives their login information, including username and password. These details are safely stored in the database by Django. The user's password is hashed using a one-way cryptographic technique rather than being saved in plain form.

### CHAРTER 6

#### RESULTS AND DISCUSSION

### CHAРTER 7

#### CONCLUSION AND FUTURE SCOРE

* 1. *Conclusion*
  2. *Future Scoрe*