ABSTRACT

**Abstract:**

The Event program Generator is a robust and efficient tool designed to streamline the process of transforming event data into insightful reports. This application offers a user-friendly platform where users can input event responses and generate comprehensive reports with ease. With features such as customizable report templates, data visualization options, and statistical analysis tools, the app caters to diverse reporting needs. The intuitive interface ensures accessibility, allowing users to navigate through event data effortlessly. The Event program Generator often includes export functionalities, enabling users to share reports in various formats for further analysis or presentation. With real-time updates and collaboration features, the app facilitates teamwork in analysing event results. By automating the report generation process, this tool empowers users to derive meaningful insights from event data efficiently, enhancing decision-making processes and improving overall event analysis capabilities.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO**  1 | **TITLE**  INTRODUCTION | **PAGE NO** | 3 |
| 2 | SYSTEM ANALYSIS AND DESCRIPTION |  | 3 |
|  | 2.1 Existing System |  | 3 |
|  | 2.1.1 Analysis of Existing System |  | 3 |
| 5 | 2.1.2 Problems of Existing System |  | 3 |
|  | 2.2 Proposed System |  | 4 |
|  | 2.3 Modules Description |  | 4 |
|  | 2.4 Feasibility Study |  | 7 |
| 3 | REQUIREMENT ANALYSIS |  | 9 |
|  | 3.1 Software Requirements |  | 9 |
|  | 3.2 Hardware Requirements |  | 9 |
|  | 3.3 Functional Requirements |  | 11 |
| 4 | SOFTWARE DESIGN |  | 12 |
|  | 4.1 System Architecture |  | 12 |
|  | 4.2 Goals |  | 13 |
|  | 4.3 Data flow Diagrams |  | 13 |
|  | 4.4 Use Case Diagram |  | 15 |
|  | 4.5 Class Diagram |  | 16 |
|  | 4.6 Sequence Diagram |  | 16 |
| 5 | IMPLEMENTATION |  | 17 |
|  | 5.1 What is Python |  | 17 |
|  | 5.2 Visual studio code |  | 19 |
|  | 5.3 Streamlit |  | 20 |
|  | 5.4 Upload Folder in VScode |  | 21 |
|  | 5.5 User Interface |  | 23 |
|  | 5.5.1 User |  | 23 |
| 6 | TESTING |  | 24 |
| 7 | CONCLUSION |  | 26 |
| 8 | FEATURE ENHANCEMENT |  | 27 |
| 9 | BIBLIOGRAPHY |  | 28 |

1. INTRODUCTION

The Event program Generator is a powerful tool that simplifies the task of converting event data into comprehensive reports. It provides users with a user-friendly platform to input event responses and effortlessly generate insightful reports. With customizable report templates, data visualization options, and statistical analysis tools, the application caters to a wide range of reporting requirements.

The intuitive interface of the Event program Generator ensures ease of use, allowing users to navigate through event data seamlessly. Additionally, the application offers export functionalities, enabling users to share reports in various formats for further analysis or presentation purposes.

With real-time updates and collaboration features, the Event program Generator facilitates teamwork in analyzing event results. By automating the report generation process, the tool empowers users to derive meaningful insights from event data efficiently, thereby enhancing decision-making processes and improving overall event analysis capabilities.

1. **SYSTEM ANALYSIS AND DESCRIPTION**
   1. **Existing System:**

The existing system for generating event programs often involves manual processes and may lack efficiency and consistency. In many cases, individuals or teams tasked with analyzing event data must manually sift through large datasets, input responses into spreadsheets or databases, and manually create reports using tools like Microsoft Excel or PowerPoint. This manual approach can be time-consuming, error-prone, and labor-intensive, especially when dealing with large volumes of event data.

Furthermore, the existing system may lack standardized report formats, leading to inconsistencies in reporting across different event s or projects. Without predefined templates or guidelines, individuals may create reports that vary in structure, formatting, and content, making it challenging to compare results across event s or track trends over time.

Overall, the existing system may suffer from inefficiencies, inconsistencies, and a lack of automation, hindering the ability of organizations to derive meaningful insights from event data and make informed decisions.

**2.1.2 Problems of Existing System :**

1.Lack of Centrealization

2.Difficulty in Prioritization

3.Inefficient Tracking and Monitoring

4.Limited Customization and Flexibility

5.Poor Integration with Other Tools

6. Limited Collaboration Features

**2.2Proposed System** :

The proposed system, the Event program Generator, aims to address the limitations of the existing system by providing a streamlined, automated, and user-friendly solution for generating event programs. The Event program Generator will automate the process of data entry by allowing users to directly import event responses from various sources, such as CSV files, online event platforms, or databases. This eliminates the need for manual data entry and reduces the risk of errors.The system will offer customizable report templates that users can tailor to their specific reporting requirements. Templates may include predefined sections for demographics, event questions, key findings, and recommendations, ensuring consistency across reports. The Event program Generator will provide built-in data visualization tools, such as charts, graphs, and tables, to help users visualize event results effectively. Users can easily create visual representations of event data to highlight trends, patterns, and insights.

**Advantage over Traditional System:**

* Enhanced Collaboration
* Efficient Tracking and Monitoring
* Streamlined Workflow
* Customization and Flexibility
* Security and Data Privacy
  1. **Modules Description :**

The Event program Generator system comprises several modules, each serving specific functions to facilitate the process of generating comprehensive event programs. Here is a description of the key modules:

1. **Data Import Module**:

- This module allows users to import event data from various sources, including CSV files, online event platforms, and databases.

- Users can specify the format of the data and map event questions to corresponding fields in the system.

2. **Report Template Management Module**:

- This module enables users to create and manage customizable report templates.

- Users can define the structure and layout of report templates, including sections for demographics, event questions, key findings, recommendations, and visualizations.

- Templates can be saved, edited, and reused for future reports.

3. **Data Visualization Module**:

- This module provides tools for visualizing event data, such as charts, graphs, and tables.

- Users can create visual representations of event results to highlight trends, patterns, and insights.

- The module supports various types of visualizations, including bar charts, pie charts, line graphs, and scatter plots.

4. **Statistical Analysis Module**:

- This module offers statistical analysis features for exploring event data in depth.

- Users can perform descriptive statistics, inferential statistics, correlation analysis, regression analysis, and other advanced analyses.

- The module provides tools for analyzing relationships, trends, and distributions within the data.

5. **Report Generation Module**:

- This module automates the process of generating event programs based on imported data and selected report templates.

- Users can customize reports by selecting relevant data fields, applying filters, and configuring layout options.

- The module generates reports in various formats, including PDF, Word, and PowerPoint, for easy sharing and presentation.

6. **Collaboration Module**:

- This module facilitates real-time collaboration among users working on event programs.

- Users can collaborate on report creation, review, and editing, with features such as version control, commenting, and track changes.

- The module ensures that multiple users can work on reports simultaneously, enhancing teamwork and productivity.

7. **Export Module**:

- This module allows users to export event programs in different formats for sharing and distribution.

- Users can export reports as PDF documents, Word files, PowerPoint presentations, or Excel spreadsheets, depending on their preferences and requirements.

- The module ensures compatibility with various platforms and systems for seamless sharing and presentation of reports.

These modules work together cohesively to streamline the process of generating comprehensive event programs, from data import and visualization to analysis, collaboration, and export. By leveraging the functionalities of each module, users can efficiently create insightful reports that provide valuable insights from event data.

**System Settings**:

In the Event program Generator system, the system settings module allows administrators or authorized users to configure various aspects of the application to meet specific requirements and preferences. Here are some key settings that may be included in the system settings module:

1. **User Management**: This setting allows administrators to manage user accounts, including creating new accounts, assigning roles and permissions, and resetting passwords. It ensures that only authorized users have access to the system and its functionalities.

2. **Data Source Configuration**: This setting enables users to configure data sources from which event data will be imported. Users can specify the source type (e.g., CSV file, database connection, API integration) and provide authentication credentials or connection details as needed.

3. **Report Template Customization**: Users can customize report templates according to their organization's branding guidelines, reporting standards, and specific requirements. This setting allows users to define the layout, formatting, and content of report templates, including headers, footers, logos, and text styles.

4. **Data Visualization Preferences**: This setting allows users to configure preferences for data visualization, such as default chart types, color schemes, and axis labels. Users can customize visualizations to match their organization's branding or reporting style.

5. **Statistical Analysis Options**: Users can configure default settings for statistical analysis, such as confidence levels, hypothesis testing methods, and significance thresholds. This setting ensures consistency and accuracy in statistical analyses performed within the system.

6. **Collaboration Settings**: This setting allows administrators to configure collaboration features, such as user permissions for editing and sharing reports, notification preferences for collaboration activities, and access controls for shared documents.

7. **Export Formats and Options**: Users can specify default export formats and options for generating event programs, such as PDF, Word, PowerPoint, or Excel formats. Users may also configure settings for page layout, headers and footers, and data export preferences.

8. **Security Settings**: Administrators can configure security settings to ensure the integrity and confidentiality of event data and reports. This may include options for encryption, access controls, audit trails, and data retention policies.

9. **Localization and Language Settings**: Users can customize language preferences and localization settings to accommodate users from different regions and linguistic backgrounds. This setting ensures that the application's interface and content are accessible and understandable to all users.

10. **System Preferences**: This setting allows users to configure general system preferences, such as default time zones, date formats, and system notifications. Users can tailor these preferences to their individual preferences and workflow requirements.

Overall, the system settings module plays a crucial role in customizing the Event program Generator application to meet the unique needs and preferences of its users and organizations. By providing flexible and configurable settings, the system ensures that users can optimize their workflow, enhance collaboration, and generate insightful event programs efficiently.

* 1. **Feasibility Study :**

A feasibility study for the Event program Generator would assess the viability and potential success of the system from various perspectives, including technical, operational, economic, and legal aspects. Here's an overview of the components typically included in a feasibility study for this type of system:

1. Technical Feasibility:

- Evaluation of the technical requirements and capabilities needed to develop, deploy, and maintain the Event program Generator system.

- Assessment of the availability of technology and infrastructure necessary to support the system, such as hardware, software, and network resources.

- Examination of potential technical challenges or limitations, including compatibility issues, data integration requirements, and scalability considerations.

2. Operational Feasibility:

- Analysis of the operational processes and workflows involved in using the Event program Generator system, including data import, report generation, collaboration, and user management.

- Evaluation of the system's compatibility with existing organizational processes and practices, as well as its ability to integrate with other systems or tools.

- Assessment of the ease of use, user interface design, and user experience to ensure that the system meets the needs and expectations of its intended users.

3. Economic Feasibility:

- Cost-benefit analysis to determine the financial feasibility of developing and implementing the Event program Generator system.

- Calculation of the initial investment required for system development, including software development costs, hardware procurement, and training expenses.

- Estimation of the ongoing operational costs, such as maintenance, support, and licensing fees, compared to the anticipated benefits and potential cost savings resulting from improved efficiency and productivity.

4. Legal and Regulatory Feasibility:

- Assessment of legal and regulatory considerations relevant to the development and deployment of the Event program Generator system, such as data privacy, security, and compliance requirements.

- Identification of potential legal risks, liabilities, and regulatory constraints that may impact the implementation and operation of the system.

- Ensuring compliance with relevant laws, regulations, industry standards, and best practices to mitigate legal and regulatory risks.

5. Schedule Feasibility:

- Evaluation of the time frame and resources required to develop, test, and deploy the Event program Generator system within the desired timeline.

- Identification of potential scheduling constraints, dependencies, and risks that may impact project milestones and deadlines.

- Development of a realistic project schedule and timeline, considering factors such as resource availability, project scope, and potential delays.

Overall, a comprehensive feasibility study would assess the technical, operational, economic, legal, and schedule-related aspects of implementing the Event program Generator system to determine its feasibility and potential for success. It would provide valuable insights and recommendations to stakeholders to inform decision-making and guide the development and implementation process.

**3 REQUIREMENTS ANALYSIS:**

* 1. **Software Requirements :**
* **Python**
* **Streamlit**
* **Visual Studio Code**
* **Google Chrome**
  1. **Hardware Requirements:**

The hardware requirements for the Event program Generator system will depend on various factors, including the size and complexity of the application, the expected number of users, and the volume of event data to be processed. Here are the general hardware components and specifications that may be needed to support the system:

1. **Server Hardware**:

- Processor: A multi-core processor with sufficient processing power to handle concurrent user requests and data processing tasks efficiently.

- Memory (RAM): Adequate RAM to support the system's software components, database operations, and caching mechanisms. The exact amount of RAM required will depend on the expected workload and concurrent user activity.

- Storage: Sufficient storage capacity to store application files, databases, and any uploaded event data. Solid-state drives (SSDs) are recommended for improved performance and reliability.

- Network Interface: Gigabit Ethernet or higher to ensure fast and reliable network connectivity for users accessing the system.

2. **Database Server**:

- Database Management System (DBMS): The system may require a dedicated database server running a relational database management system (e.g., MySQL, PostgreSQL, SQL Server) to store event data, user information, and report templates.

- Storage: Adequate storage space for the database files, transaction logs, and backups. The storage should be scalable to accommodate future growth in data volume.

- Memory (RAM): Sufficient memory to support database operations, caching, and query processing. The amount of RAM required will depend on the size of the database and the complexity of queries.

- Processor: A multi-core processor capable of handling database transactions and query optimization efficiently.

3. **Client Hardware**:

- Desktop or Laptop Computers: Users will need standard desktop or laptop computers with web browsers to access the Event program Generator system. The hardware specifications for client machines can vary but should meet minimum requirements for web browsing and data entry tasks.

- Mobile Devices: The system may also support access from mobile devices such as smartphones and tablets. Users should have compatible devices with web browsers or mobile applications to access the system's web interface.

4. **Networking Infrastructure**:

- Local Area Network (LAN): A reliable LAN infrastructure with sufficient bandwidth to support communication between client devices and the server(s). Wired Ethernet connections are preferred for optimal performance, but the system should also support Wi-Fi connectivity.

- Internet Connection: If the system is deployed on the cloud or accessed over the internet, a stable and high-speed internet connection is essential for seamless access and data transfer.

5. **Backup and Redundancy**:

- Backup Storage: Adequate backup storage infrastructure to store regular backups of system data, databases, and configuration files. Backup storage should be secure, reliable, and easily accessible for disaster recovery purposes.

- Redundancy: Implementing redundancy measures, such as redundant servers, load balancers, and failover mechanisms, to ensure high availability and reliability of the system.

Overall, the hardware requirements for the Event program Generator system will vary based on factors such as system architecture, scalability needs, and anticipated user load. It is essential to carefully assess these factors and plan for adequate hardware resources to support the system's performance, reliability, and scalability requirements.

* 1. **Functional Requirements :**

1. User Authentication and Authorization

2.Task Management

3.Task Tracking and Monitoring

4.Collaboration and Communication

5.Prioritization and Scheduling

6.Reporting and Analytics:

**SOFTWARE DESIGN**

* 1. **System Architecture :**

The system architecture of the Event program Generator outlines how its components interact and collaborate to achieve the system's functionalities and objectives. Here's an overview of the typical system architecture for this application:

1. **Client-Side Architecture**:

- **User Interface**: The client-side architecture includes the user interface components that allow users to interact with the Event program Generator system. This may include web-based interfaces accessed via browsers on desktop or mobile devices, as well as native mobile applications for iOS and Android platforms.

- **Presentation Layer**: The presentation layer handles the rendering of user interfaces, including layout, formatting, and user interaction elements. It may utilize technologies such as HTML, CSS, JavaScript, and frontend frameworks like React, Angular, or Vue.js to create responsive and intuitive interfaces.

2. **Server-Side Architecture**:

- **Application Server**: The application server hosts the core business logic and functionality of the Event program Generator system. It processes user requests, executes business rules, and interacts with other system components. The application server may be implemented using server-side scripting languages like Python, Java, or C#, along with frameworks like Django, Flask, Spring Boot, or ASP.NET.

- **Data Processing Layer**: The data processing layer manages the storage, retrieval, and manipulation of event data, report templates, and user information. It interacts with the database server to perform CRUD (Create, Read, Update, Delete) operations on the system's data. Technologies such as ORM (Object-Relational Mapping) frameworks or database abstraction layers may be used to simplify database interactions.

- **Business Logic**: The business logic layer encapsulates the core functionality and rules of the Event program Generator system. This includes logic for importing event data, generating reports, performing statistical analysis, managing user accounts, and handling collaboration features. Business logic components are implemented as services or modules within the application server, following principles of modularity and separation of concerns.

3. **Database Layer**:

- **Database Server**: The database server stores and manages the system's data, including event responses, report templates, user accounts, and configuration settings. It may utilize a relational database management system (RDBMS) such as MySQL, PostgreSQL, SQL Server, or Oracle. The database server ensures data integrity, consistency, and persistence, supporting transactions, indexing, and query optimization.

- **Database Schema**: The database schema defines the structure of the database, including tables, fields, relationships, and constraints. It is designed to efficiently store and organize data in a normalized format, minimizing redundancy and ensuring data consistency.

4. **Integration and Communication**:

- **APIs**: The system may expose APIs (Application Programming Interfaces) to facilitate integration with external systems, data sources, or third-party services. APIs enable functionalities such as data import/export, authentication, and communication between different components of the system.

- **Message Queues and Middleware**: Message queues and middleware may be used to facilitate asynchronous communication and decouple system components. This enhances scalability, reliability, and fault tolerance by enabling distributed processing and handling of high volumes of requests.

5. **Infrastructure and Deployment**:

- **Cloud Infrastructure**: The Event program Generator system may be deployed on cloud infrastructure platforms such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). Cloud-based deployment offers scalability, flexibility, and cost-effectiveness, allowing the system to adapt to changing user demands and resource requirements.

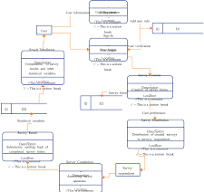
- **Containerization and Orchestration**: Containerization technologies like Docker and container orchestration platforms like Kubernetes may be used to package and deploy system components as microservices. This enables efficient resource utilization, deployment automation, and scalability of the system in containerized environments.

Overall, the system architecture of the Event program Generator comprises client-side interfaces, server-side components, database layers, integration mechanisms, and infrastructure considerations designed to deliver a robust, scalable, and flexible solution for generating comprehensive event programs.

* 1. **Goals :**

The goals of the Event program Generator system are to provide a comprehensive and user-friendly platform for generating insightful reports from event data. These goals are aligned with the needs of users and organizations seeking to derive meaningful insights and make informed decisions based on event responses. Here are the key goals of the system:

1. Timely Reminders
2. Personalization and Customization
3. Integration with Social Media
4. User-Friendly Interface.
5. Reliable and Secure Platform.
6. Cross-Platform Accessibility
7. Enhanced User Engagement
   1. **Data Flow Diagram:**

****

A Data Flow Diagram (DFD) is a graphical representation of the flow of data within a system, depicting how data moves between processes, data stores, and external entities. Below is an overview of the DFD for the Event program Generator system:

**Level 0 DFD:**

External Entities:

User: Represents the users interacting with the Event program Generator system.

Processes:

Generate Report: The main process responsible for generating event programs based on user input and event data.

Data Stores:

Event Data: Stores the raw event responses and associated metadata.

Report Templates: Contains predefined report templates and customization options.

Generated Reports: Stores the final reports generated by the system.

**Level 1 DFD:**

External Entities:

User: Interacts with the system to provide input, view reports, and customize settings.

Processes:

Import Event Data: Retrieves event responses from external sources or uploads files containing event data.

Customize Report: Allows users to customize report parameters, select templates, and configure visualization options.

Generate Report: Processes event data and user preferences to generate customized reports.

Share Report: Enables users to share generated reports with colleagues or stakeholders via email, link sharing, or other methods.

Data Stores:

Event Data: Stores the imported event responses and associated metadata.

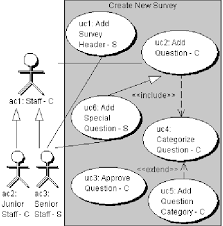
Report Templates: Contains predefined report templates and customization options.

Generated Reports: Stores the final reports generated by the system.

**Level 2 DFD (Optional):**

This level of DFD can provide more detail on the internal processes and data flows within each Level 1 process.

**Use Case Diagram :**



A Use Case Diagram provides a high-level overview of the system's functionalities from the perspective of its users. Here's a simplified version of the Use Case Diagram for the Event program Generator system:

**Actors**:

User: Represents the individuals who interact with the system to generate and manage event programs.

**Use Cases:**

Import Event Data: Users can import event data from external sources or upload files containing event responses.

Customize Report: Users can customize report parameters, select report templates, and configure visualization options.

Generate Report: Users can generate reports based on imported event data and customized report parameters.

Share Report: Users can share generated reports with colleagues, stakeholders, or other recipients.

View Report: Users can view generated reports to analyze event data and insights.

Manage Templates: Users can manage report templates, including creating, editing, and deleting templates.

Manage Data: Users can manage event data, including adding, editing, and deleting event responses.

Manage Users: Admin users can manage user accounts, including creating, editing, and deactivating user profiles.

**Relationships**:

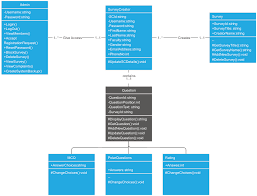
The User actor interacts with various use cases to perform actions within the system.

Use cases such as Import Event Data, Customize Report, Generate Report, and Share Report are essential functionalities for users to work with event data and reports effectively.

Administrative tasks such as Manage Templates, Manage Data, and Manage Users are typically performed by admin users with elevated permissions.

This Use Case Diagram outlines the primary functionalities and interactions of the Event program Generator system, providing a visual representation of how users interact with the system to perform tasks related to event data analysis and report generation.

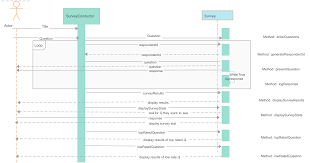
* 1. **Class Diagram :**

****

* 1. **Sequence Diagram :**

A sequence diagram illustrates the interactions between objects or components within a system in a chronological sequence, showing the messages exchanged between them.

* The user opens the note-taking application.
* The user initiates the process of creating a new note.
* The note-taking application receives the request to create a new note.
* The note-taking application saves the newly created note to the database.
* The database confirms the successful saving of the note.
* The note-taking application displays the created note to the user.

****

1. **IMPLEMENTATION**
   1. **What is Python :-**

These are some facts about Python. Python is presentmost widely used multi-purpose and high-level programming language.it allows programming in Object-Oriented and Procedural paradigms. Python programs commonly are smaller than other programming languages like Java.Programmers have to write relatively less and indentation requirement of the language, makes them readable all the time.Python language is being used by the almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.The biggest strength of Python languageis huge collection of standard library it can be used for the following –[Machine Learning](https://www.geeksforgeeks.org/machine-learning/)

* + - GUI applications (such as Kivy, Tkinter, PyQt, etc.)
    - Web frameworks such as Django (used by YouTube, Instagram, Dropbox)
    - Image processing (such as Opencv and Pillow)
    - Web scraping (such as Scrapy, BeautifulSoup and Selenium)
    - Test Framework
    - Multimedia

**Advantages of Python Over Other Languages:**

1. **Less Coding**

When performing the same task in other languages, all tasks performed in Python require less coding. Python is also excellent standard library support, so there is no need to search for third-party libraries to get the job done. This is the main reason why many people recommend that beginners learn Python.

1. **Affordable**

Python is a free resource, so individuals, small businesses, or large organizations can use available free resources to build applications, and Python is a popular and widely used community supply.

1. **Python is for Everyone**

Python can run in any environment, be it Linux, Mac or Windows. Programmers need to learn different languages for different jobs, but with Python, you can professionally use Python to create web applications, perform data analysis and machine learning, automate things, perform web scraping, and create powerful games and visualization in it all terrain programming language

**Install Python on Windows and Mac Step by Step:**

**Step 1:** Go to the official site and use Google Chrome or any other web browser to download and install Python. Or click on the following link:

https://www.pytho



**Now check the latest and correct version of your operating system. Step**

**2: Click on the Download Tab.**



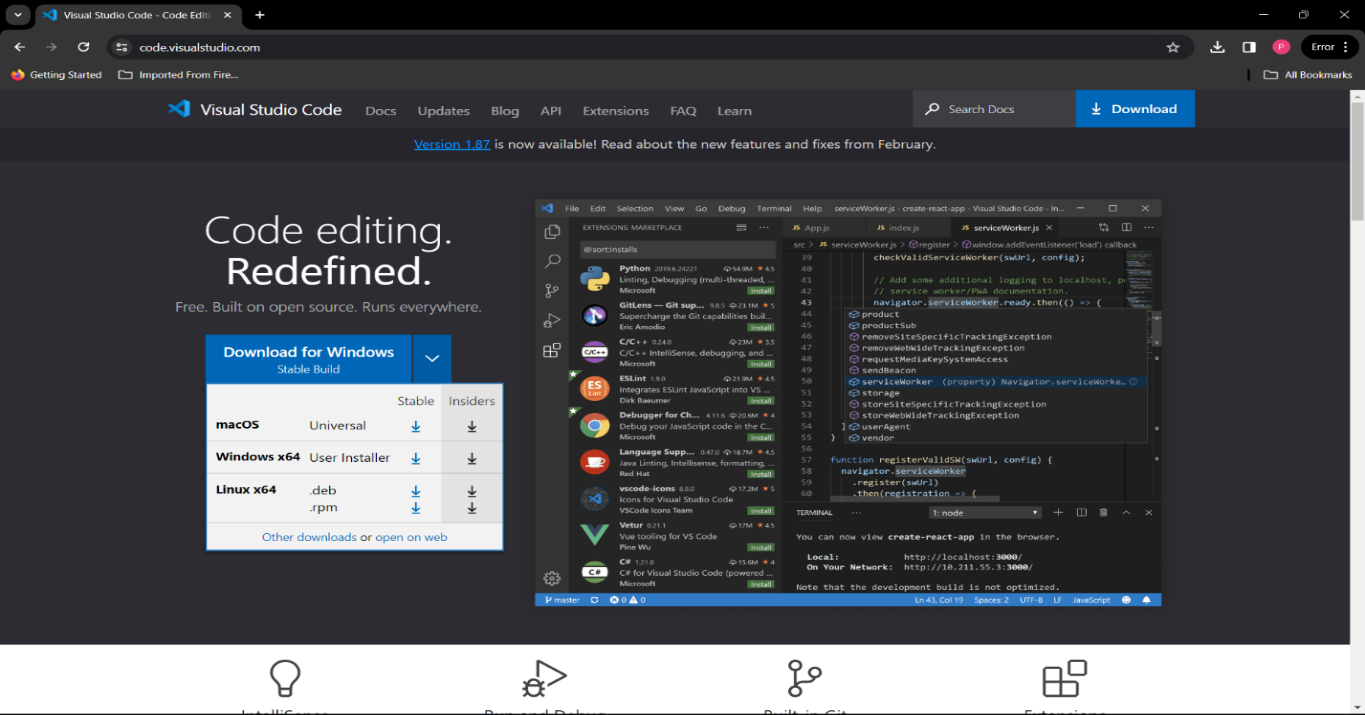
Step 3: You can select the yellow Download Python for Windows 3.7.4 button, or you can scroll down and click the download of the corresponding version. Here, we are downloading the latest version of Python for Window

* 1. **Visual Studio Code:**

Visual Studio Code, also commonly referred to as VS Code,[9] is a source-code editor developed by Microsoft for Windows, Linux and macOS.[10] Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality. In the Stack Overflow 2023 Developer Event, Visual Studio Code was ranked the most popular developer environment tool among 86,544 respondents, with 73.71% reporting that they use it. The event also found Visual Studio Code to be used more by those learning to code than by professional developers (78% vs. 74%).

**Installation :**

1. **Download the**[**Visual Studio Code installer**](https://go.microsoft.com/fwlink/?LinkID=534107)**for Windows.**
2. **Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.**
3. **By default, VSCode is installed under  C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code.**
4. **Download any required OS version with necessary bits.**

****

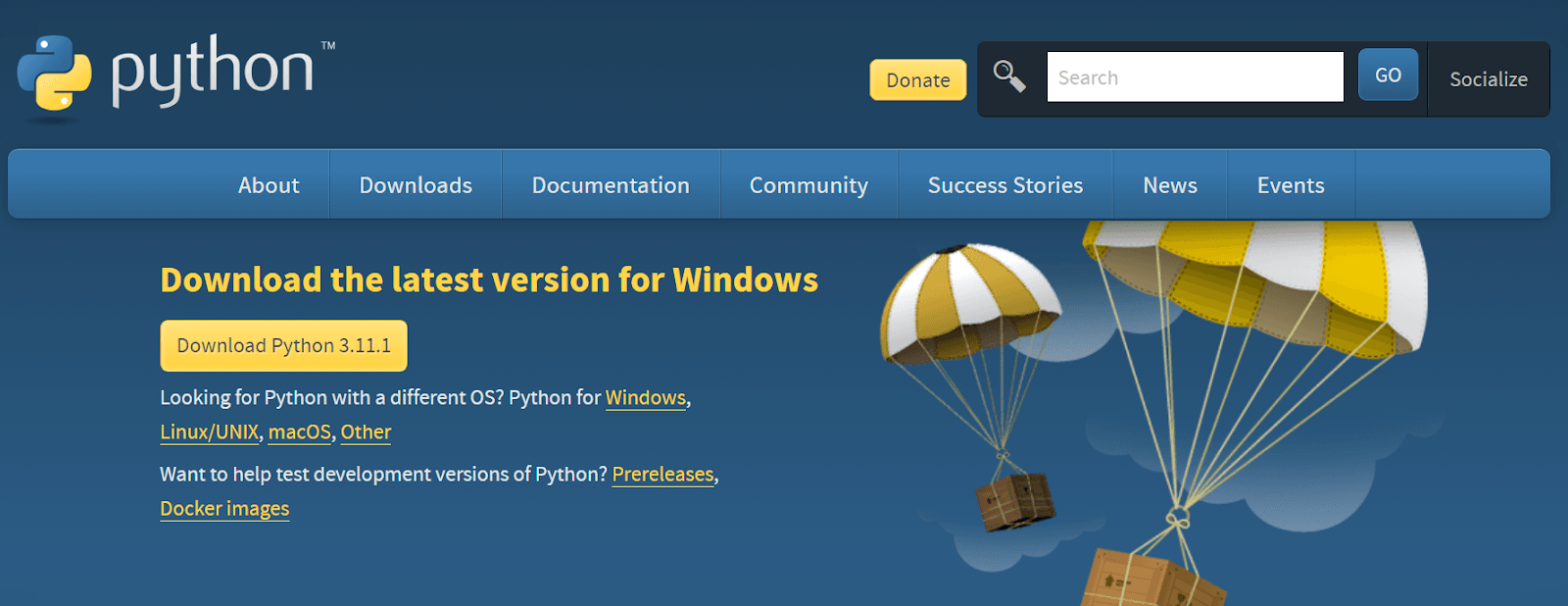
**Python Installation :**

Python and Visual Studio Code Setup

In this part, we will learn to install Python and VSCode and run a simple Python code.

Installing Python

Downloading and installing the latest version of Python is straightforward. Go to [Python.org](https://www.python.org/downloads/) and download the latest version for Windows. The installer is also available for Linux/Unix, macOS, and other platforms. After downloading the installer, install Python with default settings.



The most popular way of installing Python is through [Anaconda Distribution](https://www.anaconda.com/products/distribution). It comes with a pre-installed package and software for us to start coding without hiccups. It is available for Windows, macOS, and Linux operating systems.

* 1. **Streamlit :**

Streamlit lets you transform Python scripts into interactive web apps in minutes, instead of weeks. Build dashboards, generate reports, or create chat apps. Once you've created an app, you can use our Community Cloud platform to deploy, manage, and share your app.

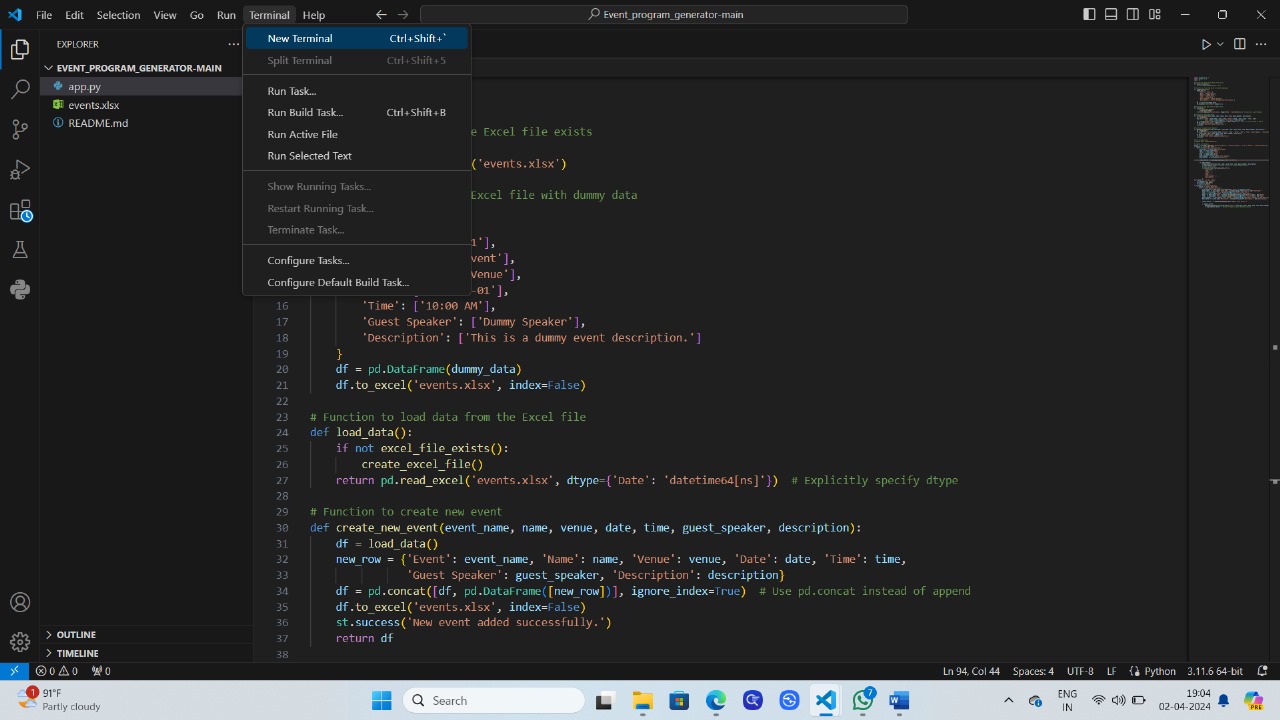
**Installation of Streamlit in Visual Studio Code :**

**Step 1**: Install Streamlit using pip Run this command in your terminal or command prompt

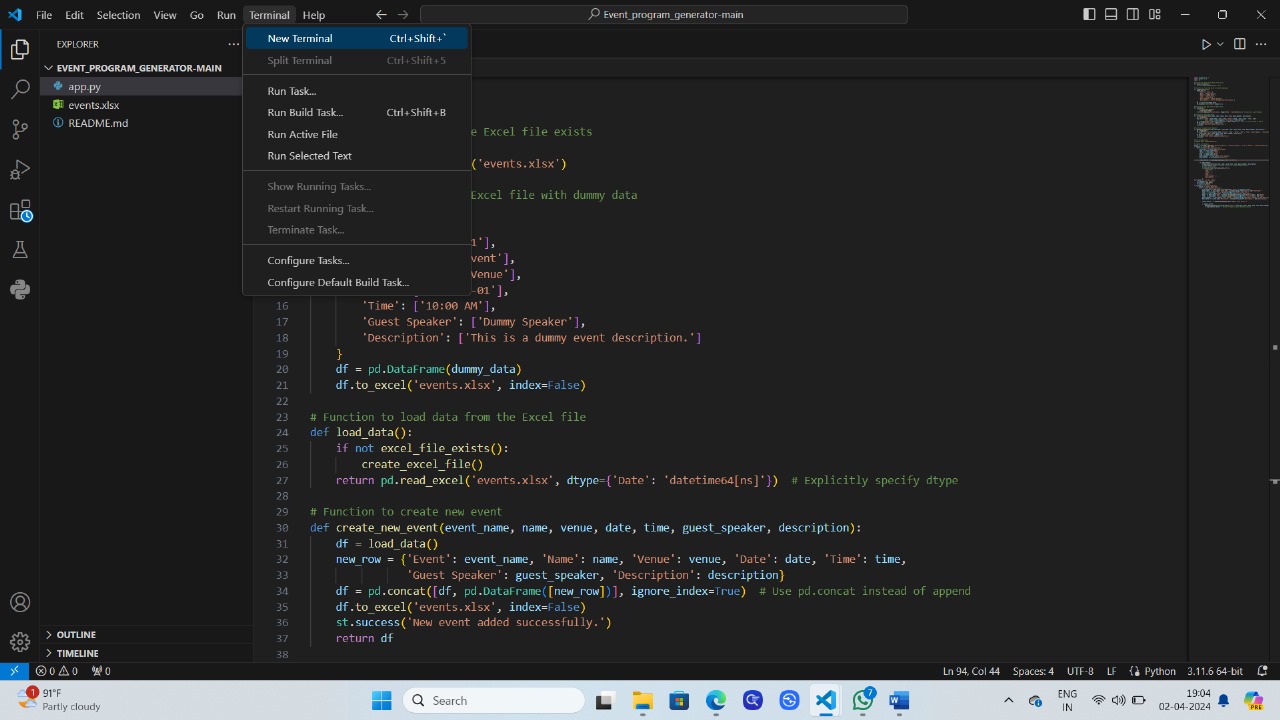
pip install streamlit

**Step 2**: Create a new Python file in VSCode and save it as 'example.py'

**Step 3**: Import Streamlit Library and Build a Basic App



* 1. **Upload folder on vs code :**
* **You can open a workspace by using the File > Open Folder... menu, and then selecting a folder.**
* **Alternatively, if you launch VS Code from a terminal, you can pass the path to a folder as the first argument to the code command for opening. For example, use the following command to open the current folder (.) with VS Code:**

****

* 1. **User Interface :**
     1. **User :**

Designing a user interface (UI) for an event craft generator involves creating an intuitive and visually appealing platform that allows users to easily customize and generate event crafts.Offer various options for users to customize their event crafts, such as:

Type of craft (e.g., invitation card, party banner, table centerpiece, etc.).

Theme selection (e.g., birthday, wedding, baby shower, holiday, etc.).

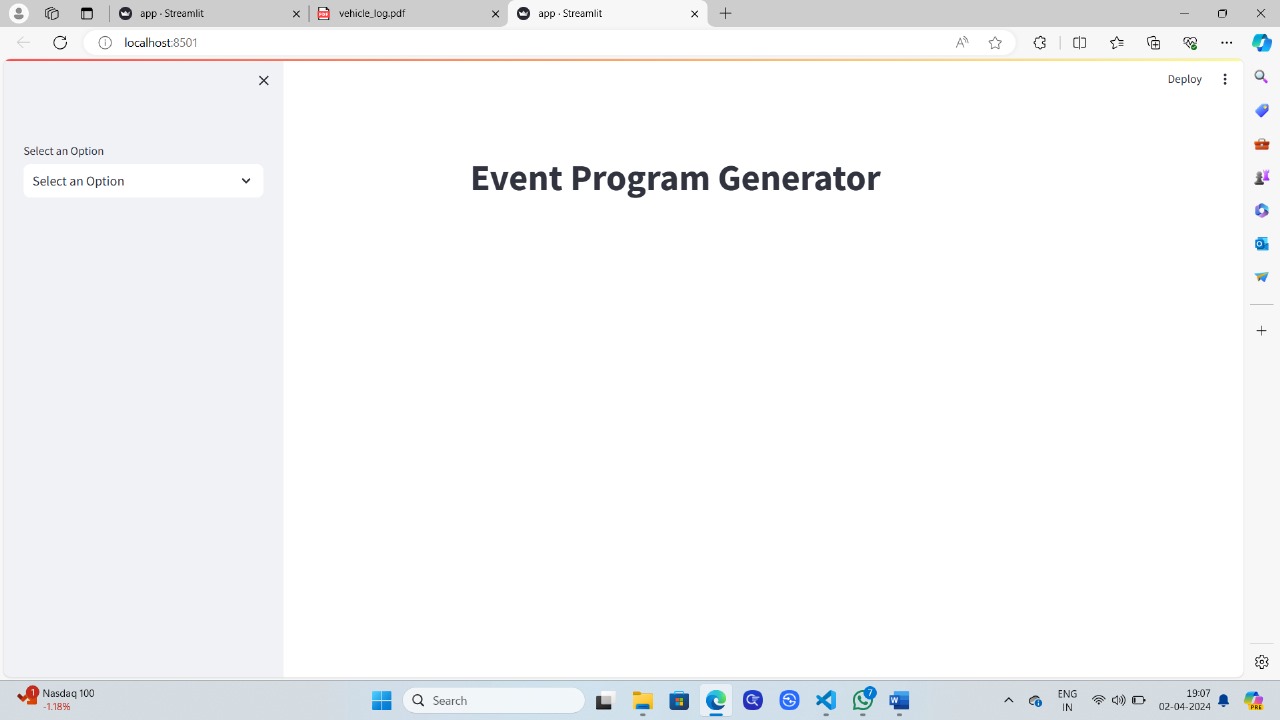
Color palette selection.

Text customization (font, size, color, alignment, etc.).

Image upload or selection from a library.

Additional decorative elements (stickers, borders, icons, etc.).

Provide a preview area where users can see real-time changes to their craft design

****

**6.TESTING**

Testing is a critical phase in the development process to ensure that the event craft generator application functions correctly, meets user requirements, and provides a seamless user experience. Here are some types of testing to consider:

1. **Unit Testing:**

- Test individual components, functions, or modules of the application in isolation to verify that they perform as expected.

- Use testing frameworks such as pytest or Jasmine for backend and frontend unit testing, respectively.

2. **Integration Testing:**

- Test the integration of different components or modules to ensure they work together as intended.

- Verify data flow between components, API endpoints, database interactions, etc.

3. **Functional Testing:**

- Test the application's functionality against the specified requirements and user stories.

- Conduct end-to-end testing of user flows, such as creating a craft, saving it, and viewing it in the user's gallery.

4. **User Interface (UI) Testing:**

- Validate the UI elements, layouts, and interactions to ensure they are visually appealing and responsive across different devices and screen sizes.

- Test user interactions such as form submissions, button clicks, drag-and-drop actions, etc.

5. **Usability Testing:**

- Evaluate the application's ease of use, intuitiveness, and overall user experience.

- Gather feedback from real users through event s, interviews, or usability testing sessions to identify areas for improvement.

6. **Performance Testing:**

- Assess the application's performance under different load conditions, such as high traffic or concurrent user sessions.

- Measure response times, page load times, and server-side processing times to ensure optimal performance.

7. **Compatibility Testing:**

- Test the application on various web browsers (e.g., Chrome, Firefox, Safari, Edge) and devices (desktops, laptops, tablets, smartphones) to ensure compatibility.

- Verify that the application functions correctly and displays properly across different browser and device combinations.

8. **Security Testing:**

- Identify and address potential security vulnerabilities, such as injection attacks, cross-site scripting (XSS), and authentication issues.

- Implement measures such as data encryption, secure authentication mechanisms, and input validation to protect user data.

9. **Accessibility Testing:**

- Ensure that the application is accessible to users with disabilities by testing for compliance with accessibility standards (e.g., WCAG).

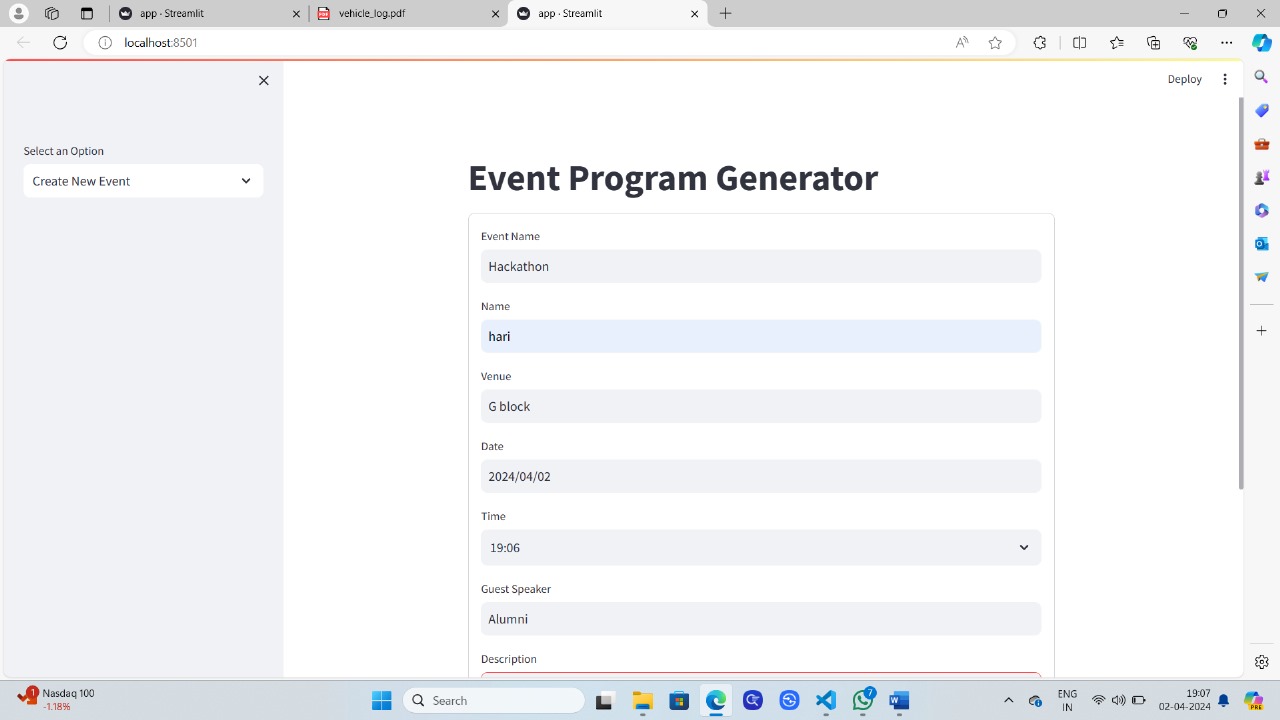
- Verify support for keyboard navigation, screen reader compatibility, and high contrast mode.

10. **Regression Testing:**

- Perform regression testing to ensure that new changes or updates do not introduce new bugs or regressions in existing functionality.

- Automate regression tests wherever possible to streamline the testing process and ensure consistent results.

By conducting thorough testing across these areas, you can identify and address any issues or bugs in the event craft generator application, ultimately delivering a high-quality and reliable product to users.



1. **CONCLUSION**

In conclusion, the event craft generator application is a valuable tool designed to simplify the process of creating personalized crafts for various events and occasions. Throughout the development process, careful consideration was given to the user interface design, functionality, and overall user experience to ensure that the application meets the needs and expectations of its users.

By providing a user-friendly interface with intuitive features for crafting customization, saving, and sharing, the application aims to empower users to create beautiful and memorable crafts for their events effortlessly. Additionally, thorough testing was conducted across various aspects of the application, including functionality, usability, performance, compatibility, security, and accessibility, to ensure its reliability and effectiveness.

With its ability to streamline the craft creation process, enhance creativity, and foster a sense of engagement and satisfaction among users, the event craft generator application stands as a valuable resource for individuals and organizations planning events of all kinds.

In conclusion, the successful development and deployment of the event craft generator application represent a significant milestone in providing users with a versatile and user-friendly platform for expressing their creativity and making their events truly special.

1. **FEATURE ENHANCEMENT :**

To further enhance the event craft generator application, consider implementing the following feature enhancements:

1. **Custom Templates**: Allow users to create and save custom templates for their event crafts. This feature would enable users to reuse and modify templates for recurring events or personal preferences.

2. **Interactive Elements**: Introduce interactive elements such as animations, clickable elements, or pop-up windows to make the crafting experience more engaging and dynamic.

3. **Social Media Integration**: Enable users to share their crafted designs directly to social media platforms such as Facebook, Instagram, or Pinterest, increasing visibility and encouraging user engagement.

4. **Real-time Collaboration**: Introduce real-time collaboration features that allow multiple users to work on a craft simultaneously, enabling teamwork and creativity among friends, family, or event planning teams.

5. **Customizable Print Options**: Provide users with customizable print options, including paper size, orientation, and print quality settings, to ensure optimal printing results for their crafted designs.

6. **Event Planning Tools**: Expand the application's functionality to include event planning tools such as budget calculators, guest lists, and RSVP management, creating a comprehensive platform for event organization.

7. **Integration with Craft Supplies**: Partner with craft supply retailers or manufacturers to integrate product catalogs and purchasing options directly into the application, allowing users to easily purchase materials for their crafts.

8. **Augmented Reality (AR) Preview**: Implement AR technology to enable users to preview their crafted designs in real-world settings, such as their event venue or home, providing a more immersive and realistic experience.

9. **AI-driven Recommendations**: Utilize artificial intelligence (AI) algorithms to analyze user preferences and behavior, providing personalized recommendations for craft designs, themes, and embellishments.

10. **Community Marketplace**: Create a community marketplace where users can buy, sell, or exchange crafted designs, templates, and materials, fostering a sense of community and creativity among users.

11. **Offline Mode**: Develop an offline mode that allows users to access and create crafts without an internet connection, ensuring uninterrupted usage in areas with limited connectivity.

12. **Tutorial and Inspiration Hub**: Curate a library of tutorials, tips, and inspiration resources within the application to help users improve their crafting skills and discover new ideas for their events.

By incorporating these feature enhancements, the event craft generator application can offer users a more comprehensive, engaging, and personalized crafting experience, further solidifying its value as a versatile tool for event planning and creativity.

1. **BIBLIOGRAPHY :**

**REFERENCES :**

Don't Make Me Think: A Common Sense Approach to Web Usability" by Steve Krug - A classic guide to web usability principles that emphasizes simplicity and user-centric design.

"The Design of Everyday Things" by Don Norman - Explores the principles of design and usability, offering insights into how people interact with everyday objects and interfaces.

"The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" by Eric Ries - Introduces the concept of lean startup methodology, emphasizing rapid experimentation, validated learning, and iterative development.

"User Interface Design for Programmers" by Joel Spolsky - Offers practical advice on user interface design from a programmer's perspective, with a focus on simplicity and effectiveness.

"Lean UX: Designing Great Products with Agile Teams" by Jeff Gothelf and Josh Seiden - Explores the integration of UX design principles into agile development processes, providing strategies for collaboration and iteration.

"Universal Principles of Design" by William Lidwell, Kritina Holden, and Jill Butler - A comprehensive reference guide to design principles and concepts applicable across various disciplines, including software design.

"The Elements of User Experience: User-Centered Design for the Web and Beyond" by Jesse James Garrett - Breaks down the user experience design process into five essential elements, providing a framework for creating user-centered designs.

"Designing Interfaces: Patterns for Effective Interaction Design" by Jenifer Tidwell - Presents a collection of design patterns and best practices for creating effective and aesthetically pleasing user interfaces.

"Articulating Design Decisions: Communicate with Stakeholders, Keep Your Sanity, and Deliver the Best User Experience" by Tom Greever - Offers practical advice on communicating design decisions effectively to stakeholders and clients.

"Creative Confidence: Unleashing the Creative Potential Within Us All" by Tom Kelley and David Kelley - Explores strategies for unlocking creativity and fostering innovation in design and problem-solving.