Event Ready!

A web application tailored towards new event planners!

Our Team

- Kevin Nguyen – nguye5tp@mail.uc.edu
- Haneesha Dushara - dusharha@mail.uc.edu
- Emilio Sese – seseev@mail.uc.edu
- Karthika Ankem – ankemka@mail.uc.edu
- Ritika Joshi - joshirk@mail.uc.edu

Our Faculty Advisor

- Dr. Richard Robles – roblesra@ucmail.uc.edu

CHERRY RED
#D90A0A

BLUE
#009CDF

GREEN
#00A259

MANGO YELLOW
#FFB000

OFF WHITE
#FFFAFA

BLACK
#000000

COLOR PALETTE

These colors reflect our brand's identity and personality

Logo & Color Scheme

Goals & Background

Project Background

- The primary purpose of this project is to provide valuable assistance to individuals who are new to event planning. We intend to empower these aspiring event planners with a user-friendly web application that streamlines the process of managing and creating events. By doing so, we aim to significantly enhance their ability to efficiently organize and budget for a wide range of events, making their entry into the world of event planning a more seamless and successful experience

Goals

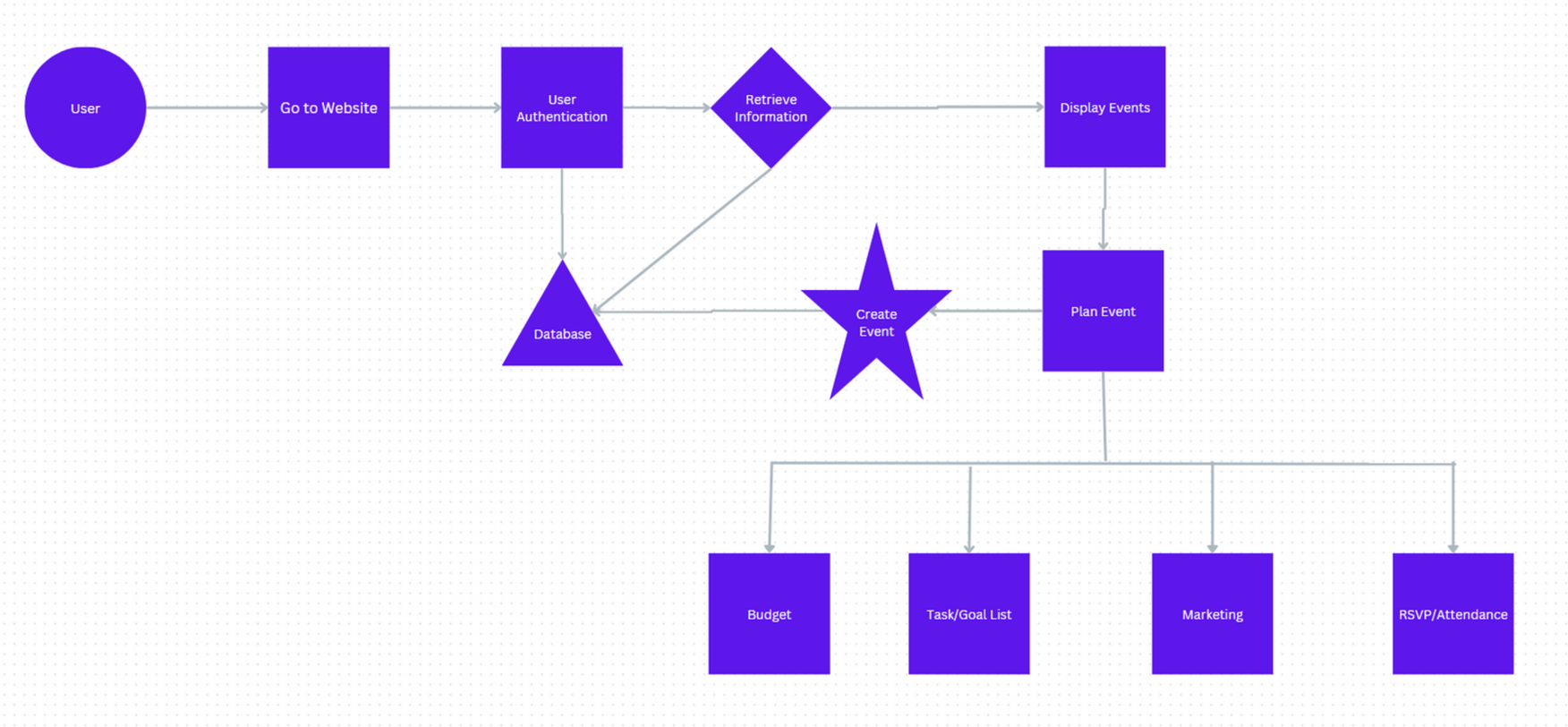
- To create a fully functional web application that effectively assists stakeholders and users in event management and planning, enabling them to efficiently organize and budget their events
- To streamline the process of event management
- Offering scheduling, budgeting, guest management, vendor coordination, and task delegation features.
- Ensuring intuitive interfaces for seamless navigation and streamlined planning.
- Providing customizable budgeting tools
- Facilitating guest list management, RSVP tracking, and communication for efficient coordination.
- Prioritizing performance and scalability for events of all sizes.
- Responsive design for accessibility across devices, enhancing on-the-go planning.

Intellectual Merits

- **Efficient Event Creation:** Utilizing Django Python and React MUI, EventReady! offers a streamlined event creation process for organizers.
- **Dynamic Marketing Tools:** Powered by Django Python and React MUI, EventReady! provides targeted marketing solutions for enhanced audience engagement.
- **Real-time Budget Monitoring:** Leveraging Django Python and React MUI, EventReady! enables organizers to monitor expenses in real-time.
- **Streamlined Task Management:** With Django Python and React MUI, EventReady! facilitates efficient task assignment and tracking for team collaboration.
- **Intuitive Design Philosophy:** EventReady! combines Django Python's backend capabilities with React MUI's interface for an intuitive user experience.

Broader Impacts

- **Student Leadership:** EventReady! empowers students to organize university events, fostering leadership skills and community engagement.
- **Vibrant Campus Life:** By simplifying event management, EventReady! enhances campus life with diverse and engaging activities.
- **Networking Opportunities:** EventReady! facilitates networking among students, faculty, and external stakeholders, fostering collaborations and connections.
- **Professional Skills:** EventReady! provides hands-on experience in event planning and budget management, preparing students for future careers.
- **Diversity and Inclusion:** EventReady! supports inclusive events that celebrate diversity, promoting cultural exchange and a sense of belonging.

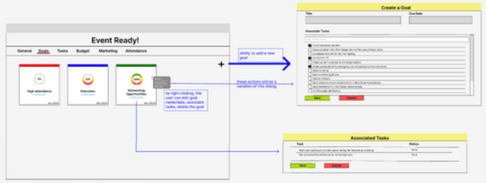


Design Diagram

LANDING PAGE

GENERAL INFO PAGE

GOALS PAGE



TASKS PAGE

MARKETING PAGE

ATTENDANCE PAGE

BUDGET PAGE

Budgeting page.

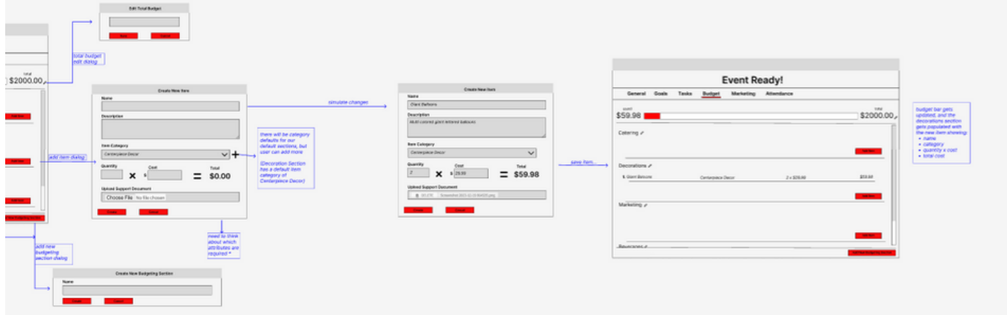
In this page, if empty, it will open a dialog to prompt the user to select categories for what their final budget might entail.
(Event Components, Decor, Food, Speakers/Performer Fees, Space Fees, etc.).

It will also ask the user their total budget (if they know that, a general estimate would work)

We will let the user create their own categories as well.

Then in these categories, the user can add items that contribute to the budget. They can specify name, quantity, cost, description, file upload, and link.

There is a total cost tracker and a live balance tracker in relation to the total budget number they initially inputted.



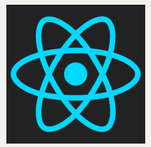
Wireframes

Technologies

- Python Django and React JS complement each other well, providing a consistent development experience from the backend to the frontend. This full-stack consistency can lead to better code organization and maintainability.
- While EventReady is a simple app, Django is scalable and can handle the growth of the application. As the project evolves, additional features can be added without significant architectural changes.
- React's component-based architecture encourages modularity and reusability of code. This is particularly advantageous for our project EventReady where various UI components have be designed and reused across different parts of the application. Material UI provides a set of pre-designed React components that adhere to Google's Material Design principles.
- SQLite is a lightweight, serverless database that is easy to set up and suitable for development purposes.

Tech Stack:

React Js, Python’s Django , SQLite3



System Architecture

Frontend (React JS, Material UI):

- React JS is employed for building modular and reusable user interface components.
- Components include those for event listing, event details, and specific pages for goals, tasks, budget, marketing, and attendance.
- Material UI is utilized for a consistent and aesthetically pleasing design.

Backend (Python Django):

- Django handles the backend logic, routing, and interacts with the database.
- Views: Implementing logic for rendering templates and handling HTTP requests.
- Models: Defining data models (e.g., Event, Task) using Django ORM for seamless database interactions.
- URLs: Routing and mapping URL patterns to views.

Database (SQLite3):

- SQLite3 is chosen for simplicity during development.
- Tables: Event, Task, Budget, Marketing, Attendance.
- Foreign keys establish relationships between tables.
- Future considerations: Mention the potential transition to a more robust database system in production.

Communication (RESTful APIs):

- RESTful APIs enable communication between the frontend and backend.
- Endpoints: Define endpoints for CRUD operations on entities like events, tasks, etc.
- HTTP methods: Properly handle GET, POST, PUT, DELETE requests.
- Data format: Use JSON for data exchange.

Milestones

Preliminaries

- **September 17th:** Surveys have been sent out to student organizations and research is being conducted
- **September 24th:** Project scope is being decided upon based upon the research
- **October 8th:** Working on UI designs and delegated research tasks
- **October 29th:** Discussed and settled on FE/BE framework and started UI/UX/Architecture Designs (React/Python-Django)
- **November 19th:** Finalized UI/UX/Architecture designs with Advisor and set up Sprints
- **December 10th:** Set up virtual environment
- **January 7th:** Delegated coding development tasks between team members

Code Development

- **[Completed] Sprint 1:** January 8th to January 21th
  - Tutorials for code implementation FE/BE
- **[Completed] Sprint 2:** January 22nd to February 4th
  - Pushed FE pages and started some BE linkage
- **[In Progress] Sprint 3:** February 5th to February 18th
- **Sprint 4:** February 19th to March 2nd
- **Sprint 5:** March 3rd to March 16th
- **Last Half-Sprint:** Development is completed on March 24th

Post Development

- **Debugging:** March 26th to April 9th
- **CEAS Expo:** April 9th @ Duke Energy Duke Energy Convention Center

Completed Results

- Gathered a lot of research from different colleges to settle on the scope of project
  - Narrowed down each feature needed to achieve our goal: *To centralize all aspects of any university event lifecycle (idea -> planning -> execution -> follow-up) in a single web application*
- Determined stack and environment
  - React Front End
  - Python-Django Back End
- Planned out tasks for each feature and delegated them accordingly to each team member
  - Two week sprints, Advisor meeting mid-sprint
- Finished programming trainings and started development
  - Landing Page FE
  - General Information Page FE / BE
  - Marketing Page FE / BE
  - Attendance Page FE

Next Steps

- Finish the rest of the code skeleton for an event
  - The rest of the FE pages
    - Goals
    - Budget
    - Tasks
  - Backend models for finished FE pages
- Feature correlation & expansion
  - Tasks to Goals
  - General Information Page card summary
  - Routing
- User authentication
  - Login page
  - Superuser and users with different permissions
- Add multiple event functionality and traversability
  - User should be able to create multiple events and each event should have its own state and respective information
- Testing, debugging, final UI completion
- The rest of the CEAS Senior Design class deliverables

Challenges

Our team of 5 students embarked on developing an app, facing initial hurdles in task allocation and learning new technologies.

Initial Challenges:

- Dividing tasks evenly among team members proved challenging.
- Learning React and Django, unfamiliar technologies for most, added complexity.

Learning Curve:

- Leveraged online resources like YouTube videos and educational documents to grasp React and Django fundamentals.
- Initial learning phase took about 2 weeks to settle in and understand project objectives.

Progress:

- Over time, gained confidence and proficiency in React and Django.
- Developed a clear vision of project objectives and technology requirements.

Achievements:

- Overcame initial challenges through perseverance and teamwork.
- Successfully applied acquired knowledge to project development, achieving significant progress.

Despite challenges, our collaborative efforts and continuous learning led to project advancement.Emphasizing adaptability and persistence as crucial factors in app development success.