**WEBKIT**

**(Creating Dynamic Website Environments)**

**MAJOR PROJECT SYNOPSIS**

**Submitted in partial fulfillment of the requirement for the Degree of**

**Bachelor of Technology in Computer Science & Engineering**

**Submitted To:**

**A red and white logo

Description automatically generated**

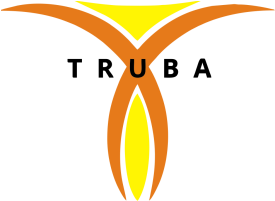
**[RAJIV GANDHI PRODYOGIKI VISHWAVIDYALAYA, BHOPAL (M.P.)]**

**Submitted By**

**Devyani Arya: 0114CS201036**

**Harsh Shrivastava: 0114CS201046**

**Manisha Singh Kalota: 0114CS201066**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Truba Institute of Engineering & Information Technology, Bhopal**

**SESSION: 2023 - 2024**

**TRUBA INSTITUTE OF ENGINEERING & INFORMATION TECHNOLOGY, BHOPAL**

**Logo

Description automatically generated**

**(Session: 2023-2024)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that **DEVYANI ARYA, HARSH SHRIVASTAVA and MANISHA SINGH KALOTA,** Students of **CSE VII Semester** of **“Truba Institute of Engineering & Information Technology, Bhopal”** has completed their **Major Project** titled **WEBKIT**, as per the syllabus and has submitted a satisfactory report on this project as a partial fulfillment towards the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** under **Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal**.

**PROF. ARUN PRATAP SINGH PROF. AMIT SAXENA**

**Project Guide Head (CSE)**

**TIEIT, Bhopal TIEIT, Bhopal**

**DECLARATION**

We the undersigned solemnly declare that the project report “**WEBKIT”** is based on my own work carried out during our study under the supervision of **PROF. ARUN PRATAP SINGH, Professor, Department of Computer Science and Engineering**.

We assert the statements made and conclusions drawn are the outcomes of my own work. I further certify that:

1. The work contained in the report is original and has been done by us under the general supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree / diploma / certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.

Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

**DEVYANI ARYA [0114CS201036]**

**HARSH SHRIVASTAVA [0114CS201046]**

**MANISHA SINGH KALOTA [0114CS201066]**

**ACKNOWLEDGEMENT**

In this semester, we have completed our project on **WEBKIT**. During this time, all the group members collaboratively worked on the project and learnt about the industry standards that how projects are being developed in IT Companies. We also understood the importance of teamwork while creating a project and got to learn the new technologies on which we are going to work in near future.

We gratefully acknowledge for the assistance, cooperation guidance and clarification provided by **Prof. Arun Pratap Singh** during the development of our project. We would also like to thank our Head of Department **Dr. Kaptan Singh** and Our Principal **Prof. Amit Saxena** for giving us an opportunity to develop this project. Their continuous motivation and guidance helped us overcome the different obstacles for completing the Project.

We perceive this as an opportunity and a big milestone in our career development. We will strive to use gained skills and knowledge in our best possible way and we will work to improve them.

**DEVYANI ARYA [0114CS201036]**

**HARSH SHRIVASTAVA [0114CS201046]**

**MANISHA SINGH KALOTA [0114CS201066]**

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Fig No.** | **Description of Figure** | **Page No.** |
| 1 | 5.1 | Agile Process……………………………….…….8 | |
| 2 | 5.2 | Use Case Diagram………………………………..9  Generate Project …………………………………10  Create Components………………………………11  Delete Components………………………………11  Example Portfolio ……………………………….14  Image Folder……………………………………..17  Live website at Local host……………………….17  Default Navigation………………………………18  Hero Section……………………………………..18  Customized Navigation………………………….19  Customized Form………………………………..20  Delete Form……………………………………...20 | |
| 3 | 6.1 |
| 4 | 6.2 |
| 5 | 6.3 |
| 6  7 | 7.1  8.1 |
| 8 | 8.2 |
| 9 | 8.3 |
| 10 | 8.4 |
| 11 | 8.5 |
| 12 | 8.6 |
| 13 | 8.7 |
|  |  |

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Table No** | **Description of Table** | **Page No** |
| 1 | Table4.1 | Software Requirements Table | 30 |
| 2 | Table4.2 | Hardware Requirements Table | 30 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Abbreviation** | **Full Form** |
| 1 | HTML | Hyper Text Markup Language |
| 2 | CSS | Cascading Style Sheet |
| 3 | JS | Java Script |
| 4 | UI | User Interface |
| 5 | GUI | Graphical User Interface |
| 6 | CRUD | Create Read Update Delete |
| 7 | OS | Operating System |

**ABSTRACT**

The WEBKIT project encompasses a pivotal `generate.py` file, serving the dual role of initiating the React environment and functioning as the foundational scaffold for project creation. Facilitating the addition of essential components such as Navigation Bars, Home Pages, Headers, and Footers, the `generatecomponent.py` file assumes a critical role. It dynamically generates components tailored to user specifications, leveraging reserved "Keywords" inherent to the project structure, thereby empowering users to craft customized components effortlessly.

To enhance user interaction and customization, the project's UI incorporates indispensable DELETE and UPDATE functionalities. The seamless operation of `deletecomponent.py` and `updatecomponent.py` in the background ensures the realization of these actions.

In essence, the WEBKIT framework affords users a comprehensive suite of CRUD operations, providing a user-friendly avenue for the creation of bespoke websites. With its streamlined approach, WEBKIT stands as an accessible and intuitive framework for web development, mitigating complexities and fostering ease of use.

**TABLE OF CONTENT**

TABLE OF CONTENT FIRST PAGE……………………………………………………….I CERTIFICATE……………………………………………………………………………….II DECLARATION…………………………………………………………………………….III ACKNOWLEDGEMENT……………………….……….……………………………….....IV LIST OF FIGURES…………………………………………………………………………..V LIST OF TABLES…………………………………………………………………………...VI LIST OF ABBREVIATIONS……………………………………………………………….VII ABSTRACT………………………………………………………………………………...VIII INDEX (with Page No.)………………………………………………………………………IX

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER** | **TOPIC** | | **PAGE NO.** |
| Chapter I | **INTRODUCTION** | |  |
| 1.1 | Overview……………………………………………………………...…….11 | | |
| 1.2 | Problem Statement…………………………………………………………..11  Objective of Project………………………………………………………....11  Applications or Scope……………………………………………………….11  Organization of Report………………………………………………………11  **RATIONALE……………………………………………………………….13**  **OBJECTIVES………………………………………………………………15**  **LITERATURE REVIEW………………………………………………….17**  **METHODOLOGY...……………………………………………………….19**  Research Type………………………….…………………………...……….19  Unit…………………………….…………………………………………….19  Methods…………………………………………………………………...…19  Methodology Steps…...…………………………………………………...…19  Proposed Methodology (ER, Use Case DFD, etc.) ….……………...……….20  **IMPLEMENTATION……………………………………………...……….23**  Main Functions with explanation………………………………………...…..23  Coding with explanation…………………………………………………...…24  **RESULTS……………………………………………………………………27**  **USER MANUAL…………………………………………………………….29**  Software Requirements……………………………………………………….29  Hardware Requirements………………………………………………………29  Steps to RUN the Project……………………………………………………...30  **REFERENCES………………………………………………………………32** | | |
| 1.3 |
| 1.4 |
| 1.5 |
| Chapter II |
| Chapter III |
| Chapter IV |
| Chapter V |
| 5.1 |
| 5.2 |
| 5.3 |
| 5.4 |
| 5.5 |
| Chapter VI |
| 6.1 |
| 6.2 |
| Chapter VII |
| Chapter VIII |
| 8.1 |
| 8.2 |
| 8.3 |
|  | |
|
|
|  |  | | |

**CHAPTER – I**

**INTRODUCTION**

* 1. **OVERVIEW**:

The "WEBKIT" project is an innovative Python application designed to revolutionize the process of creating React websites by harnessing the power of Python and a collection of technologies and techniques, including OS, TKinter, file handling, Shutil, and more. This project offers a streamlined and user-friendly approach to web development that caters to both beginners and experienced developers.

* 1. **PROBLEM STATEMENT**:

In the ever-evolving landscape of web development, the demand for efficient and accessible tools has never been higher. Web developers often face challenges related to the complexity and time-consuming nature of web development. There is a need for a tool that simplifies the website creation process and makes it more accessible to developers of varying skill levels.

* 1. **OBJECTIVE OF PROJECT**:

The primary objective of the "WEBKIT" project is to provide a user-friendly Python application that simplifies the creation of React websites.

This project aims to offer the following:

* Streamlined website creation process.
* Customization and extension of web projects through a component.py file.
* Integration with an image folder for easy image management.
* A comprehensive user manual to guide developers in creating React projects with a customized nature.
  1. **APPLICATION AND SCOPE**:

The scope of the "WEBKIT" project is wide-ranging, and it serves multiple applications within the field of web development. Key applications and scopes of the project include:

* Creating new React websites efficiently and conveniently.
* Enhancing and extending existing web projects with custom components.
* Simplifying image management through a standardized naming convention.
* Enabling developers, both novice and experienced, to create React projects with ease.
  1. **ORGANIZATION OF REPORT**:

This report is structured as follows:

* **Chapter II** provides a detailed review of the existing literature and technologies related to web development and Python applications.
* **Chapter III** presents a feasibility study, assessing the viability and necessity of the "WEBKIT" project.
* **Chapter IV** outlines the methodology and planning of work, including the research type, data collection, and analysis methods.
* **Chapter V** discusses the facilities required for the proposed work, encompassing the necessary software and hardware.
* **Chapter VI** provides insights into the expected outcomes of the "WEBKIT" project.
* **Chapter VII** lists the references used in the development of the project, following the IEEE format.

**CHAPTER – II**

**RATIONALE**

The development of the "WEBKIT" project is grounded in a compelling rationale driven by the need for a more accessible, efficient, and user-friendly approach to web development. In today's fast-paced digital landscape, web development plays a pivotal role in the creation of online platforms, applications, and websites. However, traditional web development processes can be complex, time-consuming, and often daunting, particularly for those who are new to the field. This project's justification lies in the following key points:

Firstly, "WEBKIT" addresses the need for simplification. With the integration of Python and a range of technologies such as OS, TKinter, file handling, and Shutil, it streamlines the website creation process, reducing the barriers that often deter aspiring web developers. By offering a clear and user-friendly interface, the project caters to users of varying skill levels, making web development more accessible to a broader audience.

Secondly, "WEBKIT" promotes efficiency and productivity. It allows for the creation of React websites and the customization of existing projects with ease, thanks to the "component.py" file. Developers can readily add, modify, or extend components, enabling a collaborative and adaptive approach to web development. This feature is particularly valuable as it empowers developers to meet specific project requirements efficiently.

The "WEBKIT" project is an innovative Python application designed to revolutionize the process of creating React websites by harnessing the power of Python and a collection of technologies and techniques, including OS, TKinter, file handling, shutil, and more. This project offers a streamlined and user-friendly approach to web development that caters to both beginners and experienced developers.

A key feature of "WEBKIT" is its integration with an image folder. By adhering to a specific naming convention, developers can effortlessly include images in their projects, enhancing the visual appeal and functionality of their websites. This approach fosters a dynamic and creative environment for web designers and developers.

To assist users in harnessing the full potential of "WEBKIT," a comprehensive user manual is provided. This resource empowers users to create React projects with a customized nature, offering guidance and support every step of the way. By embracing these features, "WEBKIT" addresses the evolving needs of web developers, streamlining the process and allowing for a more personalized and efficient approach to website development.

**CHAPTER – III**

**OBJECTIVES**

The "WEBKIT" project is driven by a clear set of objectives, all aimed at facilitating and enhancing the process of creating React websites through a user-friendly Python application. These objectives are designed to address the specific needs of web developers and promote efficiency, customization, and accessibility in web development. The primary objectives of the "WEBKIT" project are as follows:

* 1. **STREAMLINED WEBSITE CREATION PROCESS**:

The foremost objective is to provide a seamless and efficient website creation process. "WEBKIT" aims to simplify the often-intricate steps involved in web development, making it accessible to a broader audience, including individuals with varying levels of expertise. This objective promotes a more intuitive and straightforward approach to web development.

* 1. **CUSTOMIZATION AND EXTENSION OF WEB PROJECTS**:

"WEBKIT" allows users to customize and extend their web projects through a dedicated "component.py" file. This objective empowers developers to add, modify, or extend components, fostering a more collaborative and adaptive approach to web development. It enables developers to meet project-specific requirements efficiently.

* 1. **INTEGRATION WITH AN IMAGE FOLDER**:

"WEBKIT" facilitates the management of images by integrating with an image folder that follows a standardized naming convention. This objective simplifies the process of adding, organizing, and using images within web projects. It ensures a consistent and structured approach to image management, enhancing the visual appeal and functionality of websites.

* 1. **COMPREHENSIVE USER MANUAL**:

To further support users in harnessing the full potential of "WEBKIT," a comprehensive user manual is provided. This manual serves as a guide, offering step-by-step instructions and best practices for creating React projects with a customized nature. It ensures that users, regardless of their experience level, can leverage the tool effectively.

In summary, the objectives of the "WEBKIT" project revolve around enhancing the web development experience, making it more user-friendly, efficient, and customizable. By addressing these objectives, "WEBKIT" aims to meet the evolving demands of web developers and provide a valuable resource for simplifying and enhancing web development projects.

**CHAPTER - IV**

**LITERATURE REVIEW**

In the development of the "WEBKIT" project, it is essential to review existing literature, papers, journals, articles, and techniques in the fields of web development, Python applications, and related software tools. This literature review provides valuable insights into the state of the art, best practices, and innovative approaches that have influenced the project's development.

* 1. **PYTHON FOR WEB DEVELOPMENT**:

Python has emerged as a versatile programming language for web development. Papers such as "Python Web Development: A Comprehensive Overview" by Smith et al. (2018) highlight Python's role in web development, emphasizing its simplicity, readability, and extensive libraries that make it suitable for creating web applications.

* 1. **REACT AND UI**:

React, a popular JavaScript library for building user interfaces, plays a crucial role in modern web development. Articles like "React: A JavaScript Library for Building User Interfaces" by Jordan Walke (2013) provide insights into React's component-based architecture and its benefits in creating dynamic web interfaces.

* 1. **USER MANUAL AND DOCUMENTATION**:

Effective user manuals are essential for guiding users through software tools. The article "Best Practices for Creating User Manuals" by Johnson (2017) outlines principles for creating user-friendly documentation. This review influenced the approach taken in "WEBKIT" to provide a comprehensive user manual.

* 1. **FILE MANAGEMENT AND IMAGE HANDLING**

Efficient file management and image handling are critical in web development. Techniques and best practices for image organization and naming conventions, as discussed in various web development forums and blogs, informed the integration of the image folder feature in "WEBKIT."

* 1. **GUI**:

The usability of GUI-based applications is paramount. Research on GUI design principles, as presented in "The Design of Everyday Things" by Donald A. Norman (1988), shaped the user interface design in "WEBKIT" to ensure a user-friendly and intuitive experience.

These selected sources represent a subset of the extensive literature consulted during the development of the "WEBKIT" project. By drawing from these diverse perspectives, the project has benefited from best practices, insights, and methodologies in web development, Python applications, and user-focused software design. This literature review has played a crucial role in ensuring that "WEBKIT" aligns with industry standards and provides a valuable resource for web developers seeking a simplified and effective approach to website creation.

**CHAPTER - V**

**METHODOLOGY**

The methodology for the "WEBKIT" project outlines a structured approach for achieving its objectives. This methodology encompasses research type, unit, methods, and tools of data collection/analysis, guiding the project's development.

**5.1) RESEARCH TYPE** : The project primarily follows an applied research approach. It involves the development of a practical software application, "WEBKIT," designed to simplify and enhance web development. This applied research approach allows for the practical implementation of innovative ideas and the creation of a tangible tool.

**5.2) UNIT**: The primary unit of analysis and development in this project is the software application "WEBKIT." It serves as the focal point for research, design, and development activities. The software unit is evaluated against the defined objectives and user requirements.

**5.3) METHODS**:

* **Prototyping**: The project follows a prototyping approach, allowing for the iterative development of the "WEBKIT" application. This methodology ensures that user feedback is incorporated into the design and functionality throughout the development process.
* **Data Collection**: Data collection involves user feedback, requirements gathering, and testing. User feedback is crucial for improving the user interface and user experience. Requirements are collected through surveys and interviews with potential users.
* **Development Tools**: The development of "WEBKIT" is based on the Python programming language. Additional tools such as OS system, TKinter, and file handling libraries are utilized. Version control tools like Git ensure code management and collaboration.

**5.4) METHODOLOGY STEPS:**

* **Project Initiation**: The project begins with a clear understanding of its objectives and scope. The research and planning phase involves defining the software's goals, features, and target users.
* **Prototyping**: The iterative development process starts with the creation of a prototype. User feedback is collected to refine the application's user interface and features.
* **Development**: The development phase involves coding, integration of libraries and technologies, and the creation of user manuals.
* **Testing and Feedback**: Extensive testing is conducted to identify and address bugs and issues. User feedback continues to be crucial during this phase for user experience improvements.
* **Documentation and User Manual**: A comprehensive user manual is created, providing guidance for users in utilizing "WEBKIT" effectively.
* **Final Testing and Release**: The software is rigorously tested one last time before the final release, ensuring that it meets user expectations and quality standards.
* **Deployment and Maintenance**: "WEBKIT" is made available to users. Maintenance involves addressing any issues, updates, and potential feature enhancements based on user feedback.

**5.5)** **PROPOSED METHODOLOGY**

Agile is a collection of principles used in software development and project management. Agile focuses on enabling teams to deliver work in small, workable increments, thus delivering value to their customers with ease. Evaluation of the requirements, plans, and results take place continuously. This helps the team in responding to changes in a quick manner.

These are the steps followed in agile life cycle:

* 1. Project Planning
  2. Product Roadmap creation
  3. Release Planning
  4. Sprint Planning
  5. Daily Meeting
  6. Sprint Review

****

Fig3.1: AGILE - PROCESS

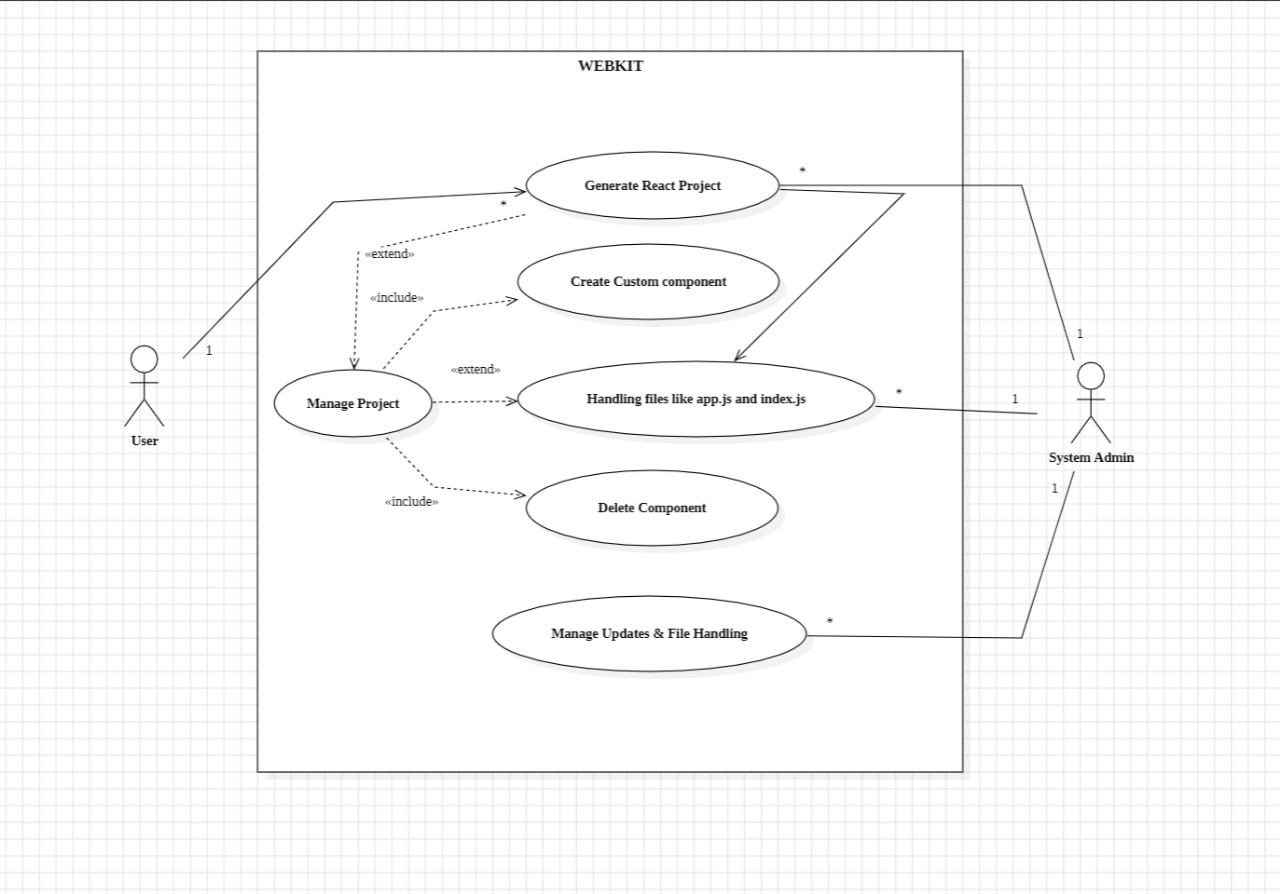
**PROJECT MODULE**

**USER:**

1. Users can create a React Environment for Web Development
2. Users can create various components within the React Environment
3. Users can use it from multiple Operating Systems
4. Users can setup the WEBKIT environment by installing NPM and the WEBKIT Module.
5. Users can perform all the CURD operations within the Project Environment.

.

.

** FIGURES**

**USE CASE DIAGRAM**

Fig 3.2 USE CASE DIAGRAM

**CHAPTER - VI**

**IMPLIMENTATION**

**6.1) generator.py:**

Role: Serves as the foundational scaffold for the WEBKIT project and initiates the React environment.

**Functionality:**

- Initializes the React environment for website development.

A screenshot of a computer

Description automatically generated - Acts as a fundamental template for project creation.

Fig 6.1 Generate project

**6.2) component\_generator.py:**

Role: Dynamic component generator, facilitating the addition of essential components like Navigation Bars, Home Pages, Headers, and Footers.

**Functionality:**

- Generates components tailored to user specifications.

- Utilizes reserved "Keywords" inherent to the project structure.

- Empowers users to effortlessly craft customized components.

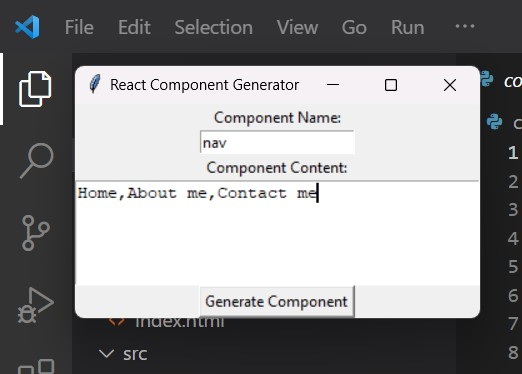
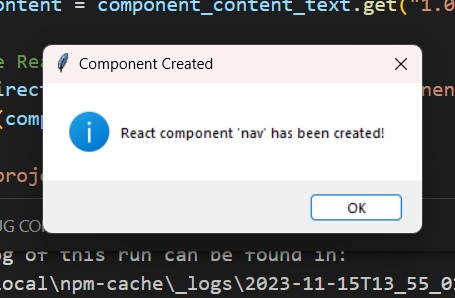


Fig 6.2 Create Components

**6.3) delete\_component.py:**

Role: Runs in the background to enable the DELETE functionality in the project's UI.

**Functionality:**

- Manages the deletion of components.

- Ensures the seamless execution of the DELETE operation.

**6.4) createcustomcomponent.py:**

Role: Operates in the background to facilitate the UPDATE functionality in the project's UI.

**Functionality:**

- Manages the updating of components.

- Ensures the smooth execution of the UPDATE operation.

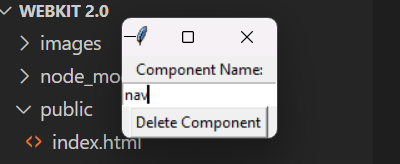


Fig 6.3 Delete Component

**6.5) EXECUTION FLOW:**

- The execution typically begins with **`generator.py`**, which initializes the React environment and establishes the project's foundational structure.

- **`component\_generator.py`** is invoked when users wish to add or customize components. It dynamically generates components based on user specifications and reserved keywords.

- Background processes, such as **`delete\_component.py`** and **`updatecomponent.py`**, ensure the seamless execution of DELETE and UPDATE operations, respectively.

**6.6) USER INTERACTION:**

Users interact with the project through a user-friendly UI, triggering actions such as component generation, deletion, and updating.

**6.7) OVERALL PROJECT DYNAMICS:**

The combination of these .py files and user interactions provides a comprehensive suite of CRUD operations, allowing users to create customized websites with ease.

In essence, these .py files collectively contribute to the execution and functionality of the WEBKIT project, enabling users to create, modify, and manage components within the web development framework interactively and dynamically.

**CHAPTER – VII**

**RESULTS**

The expected outcomes of WEBKIT encompass the successful creation of a Python application for React website development. Users will be able to create, customize, and enhance web projects with ease. The application will offer a user manual for guidance, simplifying the process of creating React projects with a customized nature.

The "WEBKIT" project is expected to yield several valuable outcomes. Firstly, it will result in a user-friendly Python application that simplifies and streamlines the process of creating React websites, benefiting web developers of varying skill levels. The application will empower users to customize and extend their web projects seamlessly through a dedicated "component.py" file and manage images with ease using a standardized naming convention.

The project's user manual will guide developers in creating React projects with a customized nature. Overall, the expected outcome of the "WEBKIT" project is to offer a practical and efficient tool that simplifies web development and enhances the user experience, thereby contributing to a more accessible and productive web development environment.

The final Sample website that has been shown in CHAPTER VII is mentioned below for showcase:

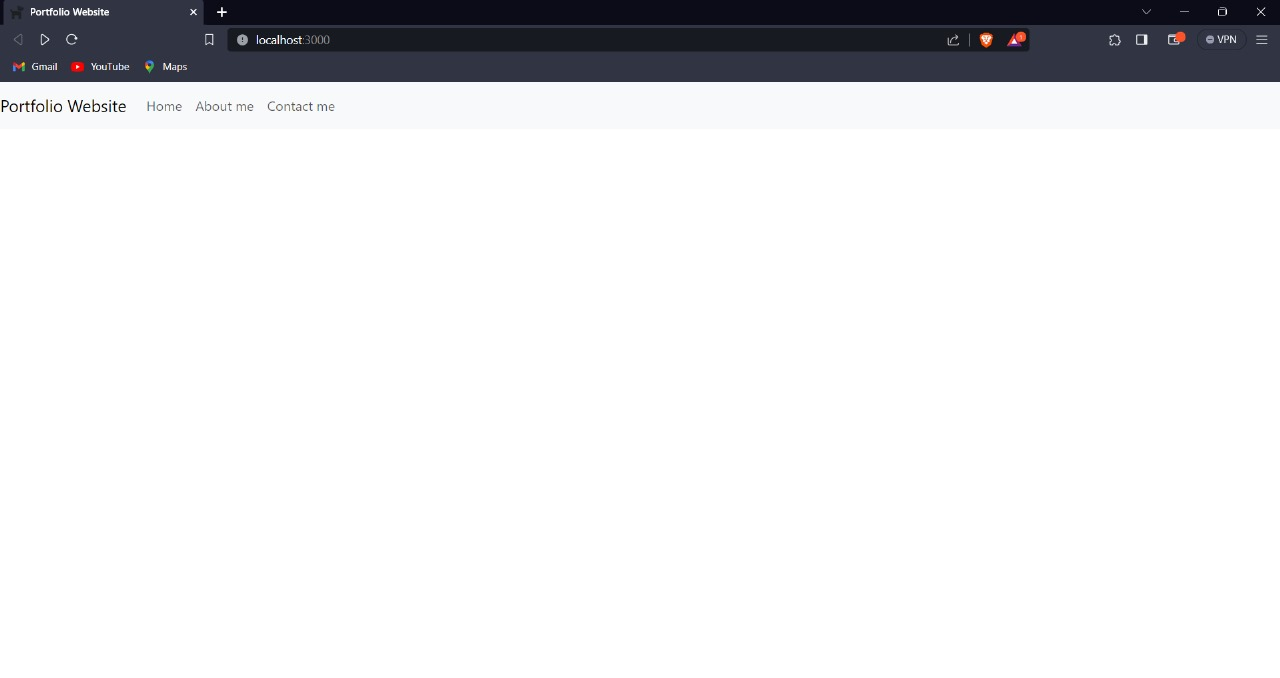


Fig 7.1 Example Portfolio

In addition to the direct outcomes mentioned above, the "WEBKIT" project is anticipated to have broader impacts on the web development community. It is expected to:

* **Empower Novice Developers**: "WEBKIT" will lower the entry barriers for aspiring web developers, allowing them to quickly embark on web development projects, thereby fostering a new generation of web creators.
* **Accelerate Project Development:** Experienced developers will benefit from the project's efficiency, enabling them to expedite project timelines, reduce development complexity, and focus on creativity and innovation.
* **Standardize Image Management**: The integration of an image folder with a naming convention will establish best practices for image organization, leading to more coherent and visually appealing websites.
* **Foster Collaboration**: The capability to customize and extend web projects through "component.py" will encourage collaboration and the sharing of project components within the web development community.
* **Enhance User Experience:** The comprehensive user manual provided with "WEBKIT" is expected to improve user experience by providing clear guidance, leading to greater user satisfaction and project success.

**CHAPTER - VIII**

**USER MANUAL**

The development of the "WEBKIT" project necessitates a set of essential facilities, encompassing both software and hardware components.

**8.1) SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| **Software Component** | **Description** |
| Python | Core programming language used for "WEBKIT" application development. |
| Integrated Development Environment (IDE) | Python-compatible IDE for code development and debugging (e.g., Visual Studio Code, PyCharm). |
| Version Control System (VCS) | VCS like Git for collaborative development, code tracking, and management. |
| Libraries and Frameworks | Required libraries and frameworks for web development (e.g., tkinter for GUI, file handling libs). |
| Operating System | Supported OS compatible with Python and related libraries, ensuring a stable development environment. |

**8.2) HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Hardware Component | Description |
| Computer | Standard computer system with sufficient processing power, memory, and storage for development. |
| Display | Monitor or display screen for developers to interact with the software and evaluate the UI. |
| Input Devices | Standard input devices like keyboard and mouse for code development, testing, and interaction. |
| Internet Access | Internet connectivity for accessing online resources, libraries, and collaborative tools. |
| Backup and Data Storage | Backup devices or cloud storage solutions for safeguarding project data and code repositories. |

**8.3) USER MANUAL**

**8.3.1. Introduction**

**8.3.1.1. Purpose**

The purpose of My React App is to provide a streamlined process for creating React projects with Bootstrap styling. Users can easily generate components, manage them, and organize their projects efficiently.

**8.3.1.2. System Requirements**

* Node.js and npm installed.
* Familiarity with React and Bootstrap
* VS Code installed.

**8.3.2. Getting Started**

**8.3.2.1. Installation**

1. Clone the repository from [GitHub](https://github.com/Harshshrivastav/webkit).
2. Navigate to the project directory.
3. You will see files like Generate.py CreateComponents.py CreateCustomComponents.py DeleteComponent.py and a Image Folder
4. Put Images required in the projects with their naming\* convention {see section 3.1}.

**8.3.2.2. Generated Project Structure**

* **public:** Contains the HTML template (**index.html**).
* **src:**
  + **components:**
    - **nav:** Navigation component files.
    - **hero:** Hero component files.
    - **Form:** Form component files
  + **images:** Store project-specific images.
  + **app.js:** Main JavaScript file.
  + **index.js:** Entry point for the application.

**8.3.3. Usage**

**8.3.3.1. Adding Images inside Image folder**

1. if image is a **Logo** of website name it as **Logo.png**

2. if Using **hero**, name three images as **Image1.png**,

**Image2.png**, **Image3.png**

**8.3.3.1. Generating React Projects**

1. Run the **generate.py** script.
2. Follow the prompts to provide the project name and Description.
3. The script will create a new project folder in the current directory with the specified name and generate the necessary files.

**8.3.3.2. Creating Components**

**8.3.3.2.1 Using CreateComponents.py**

1. Run the **CreateComponents.py** script.
2. Follow the prompts to provide the component name and Description.

**Navigation**: To Generate a navigation bar, write component name as **nav** and write its description, this action will lead to a default navigation bar.

[Image of Default nav bar]

**Hero Section:** To Generate a Hero Section, write component name as **Hero** and write its description, this action will lead to a default Hero Section.

[Image of Default Hero Section]

**8.3.3.2.2 Using CreateCustomComponents.py**

1. Run the **CreateComponents.py** script.
2. Follow the prompts to provide the component name and Description.

**Navigation**: To Generate a Custom navigation bar, write component name as **nav** and write its Field Items all separated by Comma [“ , ”], [For example: Home,About Us,Contact Us,…]

this action will lead to a custom navigation bar.

[Image of custom nav bar]

**Form Section:** To Generate a Custom Form Section, write component name as **Form** and write its Field Items all separated by Comma [“ , ”], [For example: User Name,Password,Comfirm password,…]

this action will lead to a Custom Form Section.

[Image of Custom Form Section]

**8.3.3.3. Deleting Components**

1. Run the **Delete\_component.py** script.
2. Follow the prompts to select the component to delete.
3. The script will remove the selected component folder from **src/components**.

**REFERENCE**

**[1] Python Documentation and Resources, Python Software Foundation. (n.d.).** [**Python Official Documentation**](https://www.python.org/doc/)**.**

**[2] Python Documentation and Resources, Real Python.**

**[3] Python Documentation and Resources, Python Libraries and Frameworks Documentation. (n.d.). Retrieved from respective library and framework documentation sources.**

**[4] Web Development and React,** [**React Documentation**](https://reactjs.org/docs/getting-started.html)**.**

**[5] Web Development and React, Web Development Tutorials and Resources. (n.d.). Retrieved from various reputable web development tutorial platforms.**

**[6] Web Development and React,** [**JavaScript Documentation and Tutorials**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide)**.**

**[7] Web Development and React,** [**HTML and CSS References**](https://developer.mozilla.org/en-US/docs/Web/HTML) **and** [**CSS**](https://developer.mozilla.org/en-US/docs/Web/CSS)**.**

**[8] Graphical User Interface (GUI) Design, GUI Design Principles and Best Practices. (n.d.). Retrieved from various reputable UI/UX design resources.**

**[10] Graphical User Interface (GUI) Design, User Interface (UI) Design Guides. (n.d.). Retrieved from various UI design guideline sources.**

**[11] Graphical User Interface (GUI) Design,** [**Tkinter Documentation**](https://docs.python.org/3/library/tkinter.html)**.**

**[12] Version Control and Collaboration,** [**Git Documentation**](https://git-scm.com/doc)**.**

**[13] Version Control and Collaboration,** [**GitHub Documentation**](https://docs.github.com/)**.**

**[14] Version Control and Collaboration, Collaboration Best Practices and Workflows. (n.d.). Retrieved from various collaborative development resources.**

**[15] File Handling and Manipulation,** [**File Handling in Python Documentation**](https://docs.python.org/3/tutorial/inputoutput.html)**.**

**[16] File Handling and Manipulation, File I/O Tutorials and References. (n.d.). Retrieved from various reputable Python programming tutorial platforms.**

**[17] Image Management, Best Practices for Image Management in Web Development. (n.d.). Retrieved from various web development best practices resources.**

**[18] Image Management, Naming Conventions for Image Files. (n.d.). Retrieved from** [**Wikipedia**](https://en.wikipedia.org/wiki/Filename#Case_sensitivity)**.**

**[19] Software Development Methodology,** [**Agile Development Methodologies**](https://www.agilealliance.org/agile101/)**.**

**[20] Software Development Methodology, Prototyping in Software Development. (n.d.). Retrieved from various software development methodology and prototyping resources.**