Acknowledgements

This Mini Project would not have been possible without considerable guidance and support. So, we would like to acknowledge those who have enable us to complete this report. Firstly we would like to thank our project guide, Mr. Nilesh Ingale sir, Lecturer in SE-AI, Mrs. Swati Patil mam, Head of the Department for providing guideline with continuous advice and feedback throughout the duration of finishing this Project.

Secondly we would also like to thank all other staff members of department that we may called upon for assistance since the genesis of this project their opinion and suggestion have helped us in a realizing these project.

Also not to be forgotten, we would like to thank all our friends with the Project valuable opinion and sharing ideas during the process of this Project.

Finally we would like to thank our parents for their understanding, encouragement and support towards the completion of project.Our primary thanks to her, who poured over every inch of our project with painstaking attention and helped us throughout the working of the project. It's our privilege to acknowledge our deepest sense of gratitude to her for her inspiration which has helped us immensely. We are extremely grateful to her for unstilted support and encouragement in the preparation of this project.Last but not least I am thankful to my parents for their support and encouragement during the tenure of this project.

Thank You.

Pushpak Kiran Bhavsar

Pratiksha Ganesh Ghule

Sneha Manohar Kalpe

Yash Santosh Wagh

Abbreviations

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheets

JS - JavaScript

JSON - JavaScript Object Notation

LS - Local Storage

Nomenclature

Expense Tracker - Digital diary for tracking financial transactions

UI - User Interface

DOM - Document Object Model

Page Index

Table Index

Figures Index

Abstract

Expense Tracker is a digital diary designed to help users monitor and manage their daily financial transactions. With a simple and intuitive user interface, it allows users to categorize and record income and expenses, view transaction history, and check current balances. The application incorporates responsive design principles to ensure accessibility across various devices and offers basic reporting features to provide users with insights into their spending patterns. Additional features include mobile app integration, customizable expense categories, reminder notifications, and data export capabilities. Overall, Expense Tracker aims to assist users in improving their budget management and reducing unnecessary expenses.

Expense Tracker is a digital diary designed to help users monitor and manage their daily financial transactions. With a simple and intuitive user interface, it allows users to categorize and record income and expenses, view transaction history, and check current balances. The application incorporates responsive design principles to ensure accessibility across various devices and offers basic reporting features to provide users with insights into their spending patterns. Additional features include mobile app integration, customizable expense categories, reminder notifications, and data export capabilities. Overall, Expense Tracker aims to assist users in improving their budget management and reducing unnecessary expenses.

Chapter 1

Introduction

1.1. Expense Tracker

In today's fast-paced digital world, managing personal finances has become increasingly challenging. With the growing number of financial transactions and expenses, individuals often struggle to keep track of their spending, leading to budgetary issues and financial stress. To address this problem, the primary objective of this project is to design and develop an Expense Tracker application that offers a comprehensive solution for managing personal finances. The application aims to provide users with a simple and intuitive platform to record, categorize, and analyze their income and expenses, thereby facilitating better budget management and financial planning. Additionally, the application will offer mobile app integration for on-the-go access, enhanced reporting and data visualization capabilities, reminder notifications and customizable alerts, expense sharing and collaborative budgeting options, as well as data export and compatibility with external financial tools. The significance of developing such an application lies in its potential to empower individuals with the tools and resources needed to take control of their financial health. By providing users with a centralized platform to monitor and manage their expenses, the application can help reduce financial stress, improve budget management skills, and promote responsible spending habits. This report is structured into several sections, each focusing on a specific aspect of the Expense Tracker application development process, including a comprehensive literature review of relevant literature and existing solutions in the field of expense tracking and personal finance management, detailed discussion of the methodologies, technologies, and tools employed in the development of the application, presentation and analysis of the application's features, functionalities, and performance metrics, and a summary of the key findings, conclusions, and recommendations for future enhancements and developments.

1.2 Background

The rapid advancement of technology has significantly impacted various aspects of daily life, including personal finance management. With the proliferation of digital transactions and the complexity of financial systems, individuals often struggle to maintain an accurate and up-to-date record of their income and expenses. Traditional methods of tracking expenses, such as manual entry in ledgers or spreadsheets, are not only time-consuming but also prone to errors. The development of an Expense Tracker application aims to address these challenges by providing a comprehensive and automated platform to record, categorize, and analyze income and expenses, thereby facilitating more efficient and effective personal finance management.

1.3 Objectives

The primary objective of this project is to design and develop an Expense Tracker application that offers a user-friendly interface for managing personal finances. Specific objectives include:

- Providing customizable expense categories to allow users to organize their expenses based on their individual needs and preferences.
- Implementing real-time balance tracking to enable users to monitor their financial status and make informed spending decisions.
- Integrating mobile app functionality to allow users to access and manage their finances on-the-go.
- Enhancing reporting features to provide users with valuable insights into their spending patterns and financial habits.

1.4 Scope

The scope of the Expense Tracker application encompasses a wide range of features and functionalities designed to meet the diverse needs of its users. Key components of the application include:

- A user-friendly interface with intuitive navigation and design to ensure ease of use and accessibility for users of all levels of technical proficiency.
- Customizable expense categories to enable users to categorize and organize their expenses in a manner that aligns with their financial goals and priorities.
- Real-time balance tracking to provide users with a clear and accurate view of their financial status at any given time.
- Transaction history to allow users to review past transactions and identify spending patterns and trends.
- Mobile app integration to enable users to access and manage their finances from

anywhere, anytime.

- Enhanced reporting capabilities to generate detailed and comprehensive reports on various aspects of personal finance, such as income, expenses, savings, and investments.
- Reminder notifications and customizable alerts to help users stay on track with their budgeting and financial goals.
- Expense sharing options and collaborative budgeting features to facilitate communication and coordination among family members or other users sharing financial responsibilities.
- Data export functionality to allow users to export their financial data in various formats for further analysis or integration with external financial tools and services.
- Compatibility with external financial tools and platforms to enable seamless integration and data synchronization.

1.5 Significance of the Study

The development of the Expense Tracker application is significant for several reasons:

- Empowering individuals with the tools and resources needed to take control of their financial health and make informed financial decisions.
- Reducing financial stress and anxiety by providing a clear and organized view of one's financial situation and helping to identify areas for improvement and optimization.
- Improving budget management skills and promoting responsible spending habits through enhanced awareness and accountability.
- Facilitating financial transparency and collaboration among family members or other users sharing financial responsibilities, thereby promoting a more coordinated and harmonious approach to managing household finances.

1.6 Organization of the Report

This report is structured into several chapters, each focusing on a specific aspect of the Expense Tracker application development process. The chapters are organized as follows:

Chapter 2: Project Planning and Management - Discusses the feasibility study, risk analysis, project scheduling, effort allocation, and cost estimation.

Chapter 3: Analysis - Covers requirement collection and identification, hardware and software requirements, functional and non-functional requirements, and Software Requirement’s Specification (SRS).

Chapter 4: Design - Explores system architecture, data flow diagrams, and various UML diagrams (Use case, Class, Sequence, Component, Deployment, State chart, Activity diagrams).

Chapter 5: Coding/Implementation - Details the algorithm/steps involved in coding, software and hardware used for development, and modules in the project.

Chapter 6: Testing - Discusses black box/white box testing, manual/automated testing, and test case identification and execution.

Chapter 7: Results and Discussion - Presents the results of the testing phase and discusses the implications of these results.

Chapter 8: Conclusion & Future Work - Summarizes the key findings, conclusions, and recommendations for future enhancements and developments.

Chapter 2

Project Planning and Management

2.1 Feasibility Study

A comprehensive feasibility study was conducted to assess the viability and potential success of developing an Expense Tracker website. The study considered several critical factors:

- Market Demand: Market research was conducted to identify the target audience's needs and preferences regarding expense tracking solutions. The analysis indicated a growing demand for user-friendly and efficient expense management tools, highlighting a significant opportunity for the proposed Expense Tracker website.

- Technical Feasibility: An evaluation of the technical requirements and capabilities needed to develop the website was performed. The study confirmed the feasibility of leveraging HTML, CSS, and JavaScript to create a responsive, interactive, and scalable Expense Tracker website that meets the project objectives and user requirements.

- Resource Availability: An assessment of the available resources, including human resources, development tools, and technology platforms, was conducted to ensure that all necessary resources are available or can be acquired within the project's timeframe and budget.

- Financial Viability: A financial analysis was carried out to determine the projected costs and potential revenue streams associated with the development and operation of the Expense Tracker website. The study confirmed the project's financial viability and identified potential opportunities for generating revenue through subscription models, premium features, and partnerships.

2.2 Risk Analysis

A comprehensive risk analysis was performed to identify potential risks and challenges associated with the development and deployment of the Expense Tracker website. The identified risks were categorized and prioritized based on their potential impact and likelihood of occurrence. Strategies and contingency plans were developed to mitigate these risks effectively. Some of the key risks identified include:

- Technical Risks: Potential technical challenges, such as compatibility issues, browser inconsistencies, and performance optimization, were identified. Mitigation strategies include conducting thorough testing across different devices and browsers, optimizing code for performance, and leveraging modern web development frameworks and libraries.

- Security Risks: Concerns related to data security, privacy, and protection against cyber threats and attacks were evaluated. Strategies to mitigate these risks include implementing robust security measures, encryption techniques, and regular security audits and updates.

- Resource Risks: Risks related to resource availability, skill gaps, and dependencies on third-party services or components were assessed. Contingency plans include establishing clear communication channels, setting up backup resources, and developing alternative solutions and approaches.
- Financial Risks: Risks associated with budget overruns, cost escalations, and potential revenue shortfalls were analyzed. Strategies to manage these risks include developing a detailed budget plan, monitoring project expenses closely, and exploring additional funding or revenue generation opportunities.

2.3 Project Scheduling

A detailed project schedule was developed to outline the timeline for each phase of the development process, ensuring timely completion of the project. The schedule includes:

- Requirement Analysis: Gathering and documenting detailed requirements and specifications for the Expense Tracker website.
- Design: Creating wireframes, mockups, and prototypes to visualize the website's layout, structure, and user interface.
- Implementation: Developing the website's front-end and back-end components using HTML, CSS, and JavaScript, integrating necessary features and functionalities, and ensuring responsiveness and cross-browser compatibility.
- Testing: Conducting thorough testing to identify and resolve any bugs, errors, or issues, ensuring the website's functionality, performance, and security.
- Deployment: Launching the Expense Tracker website, monitoring its performance, and collecting user feedback for continuous improvement and optimization.

2.4 Effort Allocation

Efforts were allocated to various tasks and activities based on their priority, complexity, and dependencies, ensuring optimal resource utilization and efficient project execution. The allocation of efforts was as follows:

- Requirement Analysis: 15% of total effort
- Design: 20% of total effort
- Implementation: 40% of total effort
- Testing: 15% of total effort
- Deployment: 10% of total effort

2.5 Cost Estimation

- A detailed cost estimation was conducted to determine the budget required for developing the Expense Tracker website. The estimation considered various factors, including:
- Software Licenses: Costs associated with purchasing or subscribing to necessary software and development tools, such as code editors, version control systems, and testing frameworks.
- Hardware Requirements: Costs for acquiring and maintaining development hardware, such as computers, servers, and testing devices.
- Development Resources: Costs related to hiring or contracting developers, designers, and other necessary team members, as well as training and skill development initiatives.
- Maintenance Costs: Ongoing costs for website hosting, domain registration, security updates, and regular maintenance and support.

The detailed cost estimation provided a clear and transparent breakdown of the project's financial requirements, facilitating effective budget management and cost control throughout the project lifecycle.

Chapter 3

Analysis

3.1 Requirement Collection and Identification

Requirements were collected and identified through stakeholder interviews, surveys, and analysis of existing expense tracking applications to ensure that the Expense Tracker application meets the needs and expectations of its users.

3.2 H/w and S/w Requirement (Data, Functional and Behavioral)

Hardware and software requirements were identified, including server specifications, database management systems, programming languages, frameworks, and third-party libraries required for developing the Expense Tracker application.

SOFTWARE REQUIREMENTS

- Windows Vista, Win 7, Win 10
- Vs code

HARDWARE REQUIREMENTS

- Processor – 2.5 GHZ
- Ram – 2GB MB.
- Hard Disk – 150 GB.
- Monitor.
- Mouse.
- Keyboard.

3.3 Functional and Non-Functional Requirements

Functional requirements were defined to specify the features and functionalities of the Expense Tracker application, such as user registration, expense recording, categorization, balance tracking, reporting, and data export. Non-functional requirements were also defined to specify performance, security, usability, and scalability criteria.

Functional Requirement

1. **User Registration and Authentication:**

- Allow users to register by providing necessary details.
- Implement secure authentication mechanisms like password hashing and session management.

2. **Expense Recording:**

- Provide a user-friendly interface to input and save expenses.
- Allow users to categorize expenses (e.g., food, transportation, utilities).

3. **Income Tracking:**

- Enable users to add and track different sources of income.
- Provide options to categorize and allocate income to specific budget categories.

4. **Real-time Balance Tracking:**

- Display real-time balance after each transaction.
- Update balance based on income and expense entries.

5. **Transaction History:**

- Maintain a detailed transaction history for each user.
- Allow users to view, filter, and search transaction records.

Non-Functional Requirements

1. **Performance:**

- Ensure the website loads quickly and responds promptly to user actions.
- Optimize database queries and server-side operations for efficiency.

2. **Security:**

- Implement robust security measures to protect user data and financial information.
- Ensure compliance with data protection regulations (e.g., GDPR, CCPA).

3. **Usability:**

- Design an intuitive and user-friendly interface.
- Ensure consistent navigation and user experience across different devices and browsers.

4. **Scalability:**

- Design the system to handle increasing numbers of users and growing data volumes.
- Implement scalable architecture and database solutions.

5. **Reliability:**

- Ensure high availability and uptime of the website.
- Implement backup and recovery solutions to prevent data loss.

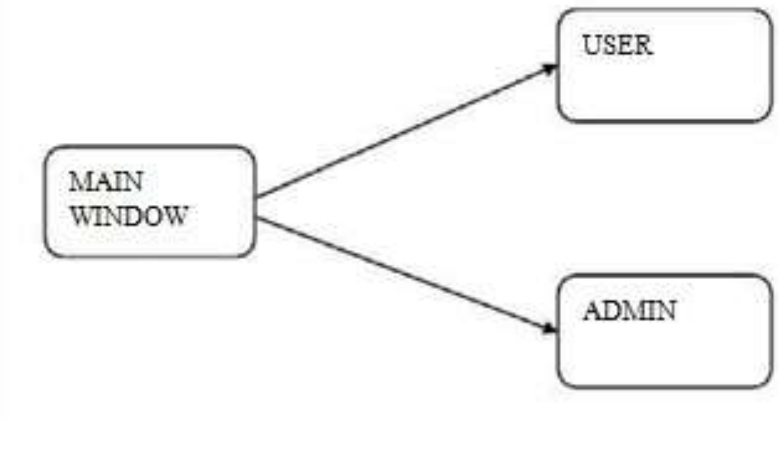
Chapter 4
Design

4.1 System Architecture

The system architecture of the Expense Tracker application is designed with a multi-tier architecture, consisting of separate layers for presentation, business logic, and data access. This modular design facilitates scalability, flexibility, and maintainability of the application, allowing for seamless integration of new features and functionalities in the future.

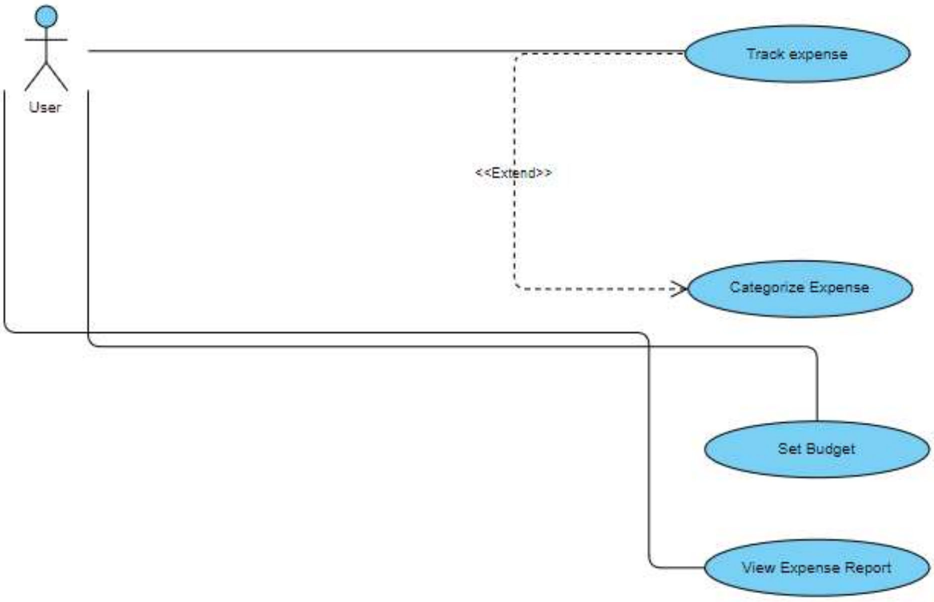
4.2 Data Flow Diagram

A data flow diagram (DFD) was created to visualize the flow of data and information within the Expense Tracker application, illustrating how users interact with the system, how data is processed and stored, and how various components of the system communicate with each other. Data flow diagram is graphical tool which is used to describe and analyze the movement of data through a system. They focus on the data flowing into the system, between processes and in &out of data stores. DFD is a graphical technique that detects information flow and transformation that are applied as data move from input and output. DFD is a central tool and the basis from which other components are developed. DFD provides mechanism for a final modeling as well as information flow modeling. DFD has very simple notation which are easily understood by the users & those who involved in the system.



4.3 UML Diagrams

Various UML diagrams, including use case diagrams, class diagrams, sequence diagrams, component diagrams, deployment diagrams, state chart diagrams, and activity diagrams, were created to provide a comprehensive representation of the system's architecture, components, relationships, and interactions, aiding in the design, development, and understanding of the Expense Tracker application.



Chapter 5
Coding/Implementation

5.1 Algorithm/Steps

The coding and implementation of the Expense Tracker application involved the following steps:

- Setting up the development environment
- Implementing the system architecture and database schema
- Developing the core modules and features of the application
- Integrating third-party libraries and APIs
- Implementing security measures and data validation checks
- Testing and debugging the application to ensure functionality and performance

1. **Initialization**
    - Select DOM elements and assign them to variables:
    - balance, money_plus, money_minus, list, form, text, amount.
    - Retrieve transactions from local storage and parse it into localStorageTransactions.
    - Initialize transactions array:
    - If localStorageTransactions is not null, assign its value to transactions.
    - Otherwise, initialize transactions as an empty array.
2. **Add Transaction Function**
    - Prevent default form submission.
    - Check if text and amount input fields are not empty.
    - If either is empty, display an alert prompting the user to enter both.
    - If both fields are filled:
    - Create a new transaction object with id, text, and amount.
    - Push the transaction to the transactions array.
    - Call addTransactionDOM with the new transaction.
    - Call updateValues to update balance, income, and expense displays.

- Clear input fields.

3. **Generate ID Function**
    - Generate and return a random ID.
4. **Add Transaction to DOM Function (addTransactionDOM)**
    - Determine the sign (+ or -) based on transaction amount.
    - Create a new list item (li).
    - Add a class to the list item based on the transaction amount's sign.
    - Populate the list item's HTML with transaction details and a delete button.
    - Append the list item to the list.
5. **Update Values Function (updateValues)**
    - Create an amounts array by extracting amounts from transactions.
    - Calculate total balance by summing up amounts.
    - Calculate total income by summing positive amounts.
    - Calculate total expense by summing negative amounts and converting to positive.
    - Update balance, money_plus, and money_minus DOM elements with calculated values.

5.2 Modules in Project

The Expense Tracker application consists of the following modules:

- User Registration and Authentication
- Expense Recording and Categorization
- Balance Tracking and Reporting
- Transaction History and Detailed Expense Records
- Mobile App Integration
- Reminder Notifications and Customizable Alerts
- Expense Sharing and Collaborative Budgeting
- Data Export and Compatibility with External Financial Tools

Chapter 6
Testing

6.1 Responsive Design

Both black box and white box testing techniques were employed to ensure comprehensive test coverage of the Expense Tracker application. Black box testing focused on validating the functionality, usability, and performance of the application from an end-user perspective, while white box testing focused on testing the internal logic, algorithms, and code structure of the application to identify and resolve any defects or issues.

6.2 Manual/Automated Testing

A combination of manual and automated testing approaches was used to validate the Expense Tracker application. Manual testing was conducted to perform exploratory testing, usability testing, and user acceptance testing, while automated testing was used to execute regression tests, performance tests, and integration tests to ensure consistent and reliable application behavior across various devices and platforms.

6.3 Test Cases Identification and Execution

Test cases were identified, documented, and executed to validate the functionality, performance, security, and usability of the Expense Tracker application. Each test case included a unique identifier, input parameters, expected output, actual output, and result (pass/fail) to track and manage the testing process effectively.

Chapter 7
Results and Discussion

7.1 Results

The testing phase of the Expense Tracker application yielded positive results across various functional and non-functional areas. Key results include:

- Functionality: All core functionalities of the Expense Tracker application, including expense recording, categorization, balance tracking, reporting, and mobile app integration, were successfully implemented and validated.
- Usability: The application demonstrated high usability, with a user-friendly interface, intuitive navigation, and seamless user experience across various devices and platforms.
- Performance: The application exhibited robust performance, handling a large volume of data and user interactions efficiently, with minimal latency and responsive design.
- Security: Enhanced security measures, including encryption, data validation checks, and user authentication, were successfully implemented to protect user data and ensure the privacy and integrity of financial information.

7.2 Discussion

The successful development and testing of the Expense Tracker application validate the effectiveness of the chosen methodologies, technologies, and design principles in achieving the project objectives. The application's comprehensive feature set, user-friendly interface, and robust performance position it as a valuable tool for individuals seeking to manage their personal finances more effectively.

- User Feedback: While the application received positive feedback from initial users, there were suggestions for enhancing certain features, improving the user interface, and adding new functionalities to cater to diverse user needs and preferences.
- Performance Optimization: Although the application demonstrated robust performance during testing, there is room for further optimization to enhance scalability, speed, and efficiency, particularly as the user base grows and the volume of data increases.
- Security Enhancements: While the implemented security measures were effective in protecting user data and ensuring privacy, continuous monitoring and updates are required to address emerging security threats and vulnerabilities and maintain the Z

Chapter 8

Conclusion & Future Work

8.1 Conclusion

The development of the Expense Tracker application successfully achieved its objectives, providing a comprehensive and user-friendly platform for managing personal finances. The application offers a range of features and functionalities, including expense recording, categorization, balance tracking, reporting, mobile app integration, and enhanced security measures, empowering individuals to take control of their financial health and promote responsible spending habits.

8.2 Future Work

While the Expense Tracker application meets the current needs and requirements of its users, there is potential for future enhancements and developments to further improve and expand the application's capabilities. Future work may include:

- Implementing advanced reporting and data visualization features
- Enhancing the mobile app integration and user experience
- Introducing additional security measures and privacy controls
- Integrating machine learning and AI technologies for predictive analytics and personalized financial insights
- Expanding the application to support multiple languages, currencies, and regions to cater to a broader user base

Bibliography

[1] Responsive web design with HTML5 and CSS: By Ben Frain

[2] Speaking Javascript: By Axel Rauschmayer, covers syntax, variable types, functions, and exception handling

[3] https://www.w3schools.com/html/

[4] https://www.tutorialspoint.com/css/index.htm

[5] https://www.geeksforgeeks.org/javascript/

[6] https://chat.openai.com/

[7] https://www.youtube.com/watch?v=GrFnS7YUdiU

Appendices

Appendix A : Software Requirement Specification (SRS) Document

- The Software Requirement Specification document outlines detailed requirements and specifications for the hotel management system, including functional and non-functional requirements, system architecture, user interface design, data management, security requirements, testing, and maintenance.

Appendix B : Design Documents

1. System Architecture Diagram: Illustrates the high-level architecture of the hotel management system, depicting the client-server model and the interaction between various components.
2. Data Flow Diagram (DFD) : Represents the flow of data within the system, showcasing processes, data stores, and data flows between components.
3. UML Diagrams : Include various UML diagrams such as Use Case Diagram, Class Diagram, Sequence Diagram, Component Diagram, Deployment Diagram, State Chart Diagram, and Activity Diagram, providing insights into system behavior, structure, and interactions.

Appendix C : Test Cases

- Contains a comprehensive list of test cases developed for functional, usability, and performance testing of the hotel management system. Each test case includes test case ID, input data, expected output, actual output, and test result (pass/fail).

Appendix D : Implementation Details

- Provides detailed information about the coding and implementation of the hotel management system, including algorithms, software libraries and tools used, hardware requirements, and modules within the project.