|  |
| --- |
| **A**  **PROJECT REPORT ON** |
|  |
|  |
| The Online Book Store Management System |
|  |
|  |
| SUBMITTED IN  PARTIAL FULFILLMENT OF  **DIPLOMA IN ADVANCED COMPUTING (PG-DAC)** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **BY**  **ABHIJEET TRIPATHI** |
|  |
|  |
| **UNDER THE GUIDENCE OF**  **Rajiv Kamune** |
|  |
|  |
|  |
| **AT**  **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE** |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE.** | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
| **CERTIFICATE** | | | |
|  | | | |
| This is to certify that the project | | | |
|  | | | |
| The Online Book Store Management System | | | |
|  | | | |
| Has been submitted by | | | |
|  | | | |
| **ASEEM VAIRAGI** | | | |
|  | | | |
|  | | | |
| In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC Sep 2023)** as prescribed by The **CDAC** ACTS, PUNE. | | | |
|  | | | |
|  | | | |
| Place: Pune | | | Date: 22-FEB-2024 |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | **Rajiv Kamune** |
|  | | | **Project Guide** |
|  |

**ACKNOWLEDGEMENT**

I would like to express my special thanks of gratitude to my teacher Rajiv Kamune as well as our Director Nitin Khudale who gave me the golden opportunity to do this wonderful project on the topic “The Online Book Store Management System”, which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Aseem Vairagi

**ABSTRACT**

The Online Book Store Management System is a comprehensive software solution designed to streamline the operations of bookstores in the digital age. With the increasing demand for online shopping, traditional brick-and-mortar bookstores need to adapt to the digital landscape to remain competitive. This system provides an efficient platform for both customers and bookstore administrators to interact and manage various aspects of book selling and purchasing.

|  |  |  |
| --- | --- | --- |
|  | **INTRODUCTION** | 1 |
|  | 1.1 Introduction | 2 |
|  | **Product Overview and Summary** |  |
|  | 2.1 Purpose |  |
|  | 2.2 Scope |  |
|  | 2.3 User Classes and Characteristics |  |
|  | 2.4 Design and Implementation Constraints |  |
|  | **REQUIREMENTS** |  |
|  | 3.1 Functional Requirements |  |
|  | 3.1.1 Use case for Administrator. |  |
|  | 3.1.2 Use case for User. |  |
|  | 3.2 Non - Functional Requirements |  |
|  | 3.2.1 Performance Requirement |  |
|  | 3.2.2 Reliability Requirement |  |
|  | 3.2.3 Security Requirement |  |
|  | 3.2.4 Scalability Requirement |  |
|  | 3.2.5 Usability Requirement |  |
|  | 3.2.6 Compatibility Requirement |  |
|  | 3.2.7 Maintainability Requirement |  |
|  | **PROJECT DESIGN** |  |
|  | 4.1 Data Model |  |
|  | 4.1.1 Database Design |  |
|  | 4.2 Process Model |  |
|  | 4.2.1 ERDiagram |  |
|  | **PROJECT SCREENSHOTS** |  |
|  | **CONCLUSION** |  |
|  | **Future Work** |  |

**INDEX**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Table Title** | **Page** |
| **Fig 1** | **Complete Database** | **19** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Section** | **Figure Title** | **Page** |
| **Fig 1** | **Use Case For User** | **15** |
| **Fig 2** | **Use Case For Admin** | **15** |
| **Fig 3** | **Complete Database** | **19** |
| **Fig 4** | **ER Diagram** | **20** |

**INTRODUCTION**

The Online Book Store Management System is a comprehensive software solution designed to address the needs of both customers and bookstore administrators. It provides a user-friendly interface that enables customers to explore a wide range of books, make purchases securely, and enjoy personalized recommendations tailored to their preferences. For bookstore administrators, the system offers powerful tools for managing inventory, tracking sales, and analyzing customer data to optimize business operations.

This introduction aims to provide an overview of the key functionalities and benefits of the Online Book Store Management System, highlighting its role in helping bookstores stay competitive and relevant in an increasingly digital world. Through its intuitive interface, advanced features, and robust management capabilities, the system empowers bookstores to meet the demands of modern consumers while maximizing efficiency and profitability.

Top of Form

Top of Form

Top of Form

**The goal of this project:**

* **Enhanced User Interface:** Utilizing React on the frontend enables the creation of a dynamic and intuitive user interface. Customers will benefit from features such as real-time search, interactive book listings, and seamless navigation, resulting in an engaging and user-friendly shopping experience.
* **Efficient Backend Development:** Spring Boot provides a powerful and streamlined framework for building the backend infrastructure of the system. Leveraging Spring Boot's auto-configuration and convention-over-configuration approach, developers can rapidly develop robust backend services, including RESTful APIs for managing inventory, processing orders, and generating reports.
* **Responsive Design:** The combination of React's component-based architecture and Spring Boot's RESTful services enables the creation of a responsive and adaptable web application that delivers a seamless experience across various devices and screen sizes.

**Product Overview and Summary**

**| Purpose:** The purpose of developing an Online Book Store Management System using Spring Boot and React is to address the evolving needs of bookstores in the digital age and provide them with a robust, efficient, and user-friendly platform to thrive in the competitive online marketplace.

**| Scope**: This project aims to create a simplified version of an Online Book Store Management System using only Spring Boot for the backend and React for the frontend. The streamlined scope includes:

1. **Backend Development with Spring Boot:**

* Designing and implementing backend functionalities using Spring Boot.
* Developing RESTful APIs for basic operations such as book listing, search, and user authentication.
* Utilizing in-memory databases like H2 for simplicity in data storage and retrieval.

1. **Frontend Development with React:**

* Creating a minimalistic user interface using React components.
* Designing UI elements for browsing books, searching, and displaying book details.
* Implementing basic navigation and user interaction features.

1. **User Authentication:**

* Implementing basic user authentication using Spring Security.
* Providing login and registration functionalities with simple validation.

1. **Inventory Management:**

* Developing features for basic book listing management.
* Implementing functionalities for adding, updating, and removing books.

1. **Order Processing:**

* Implementing basic order processing workflows for placing and viewing orders.
* No integration with payment gateways; orders are considered as placeholders.

1. **Deployment and Testing:**

* Setting up a local development environment for testing.
* Conducting manual testing to ensure basic functionality and usability.
* No automated testing or deployment processes are implemented due to the simplified nature of the project.

1. **Documentation and Support:**

* Providing minimal documentation on setting up and running the application.
* Offering basic support resources for developers to understand the project structure and implementation details.

The project focuses on delivering a minimal viable product (MVP) of an Online Book Store Management System using only Spring Boot and React. It aims to provide essential functionalities for basic book management and user interaction while keeping the project simple and easy to understand for developers. Additional features and integrations beyond the defined scope may be considered for future iterations based on project requirements and constraints.

**| User Classes and Characteristics**:

1. **Customers:**

* Characteristics:

1. Simplify the registration and login process to minimize friction for customers.
2. Streamline the browsing and searching experience to help customers find books quickly.
3. Implement personalized recommendations based on browsing and purchase history to enhance the shopping experience.

* Roles and Actions:
  + 1. Focus on intuitive and user-friendly interfaces for browsing, searching, and purchasing books.
    2. Provide clear and concise product information, including summaries, reviews, and ratings.
    3. Optimize the checkout process for ease of use and seamless transaction completion.

1. **Administrators:**

* Characteristics:
  + 1. Provide a dashboard with key metrics and insights to help administrators monitor sales performance, inventory levels, and user activity.
    2. Simplify user management functionalities, such as adding or removing user accounts and adjusting permissions.
    3. Implement automated alerts or notifications for inventory replenishment, order processing, and system errors.
* Roles and Actions:
  + 1. Prioritize efficiency in backend tools and interfaces for managing inventory, processing orders, and generating reports.
    2. Enable batch operations and bulk editing capabilities to streamline repetitive tasks.
    3. Integrate with third-party services or APIs for additional functionalities, such as shipping and logistics management.

1. **Developers:**

* Characteristics:
  + 1. Provide comprehensive documentation and resources for developers to understand the system architecture, codebase, and implementation details.
    2. Implement modular and extensible codebase architecture to facilitate future enhancements and modifications.
    3. Foster a collaborative development environment with clear communication channels and agile development practices.
* Roles and Actions:
  + 1. Prioritize clean and maintainable code practices to ensure the scalability and long-term viability of the system.
    2. Implement automated testing and continuous integration practices to maintain code quality and minimize errors.
    3. Monitor system performance and user feedback to identify areas for optimization and improvement.

**| Design and Implementation Constraints**

1. **Technology Limitations:**
   * + Constraints: The project is restricted to utilizing Spring Boot for backend development and React for frontend development, limiting the choice of additional frameworks or libraries.
     + Implications: Developers must maximize the capabilities of Spring Boot and React to meet project requirements without relying on external dependencies, ensuring compatibility and minimizing complexity.
2. **Resource Constraints:**
   * Constraints: Limited server capacity and bandwidth may impact system performance and scalability.
   * Implications: Developers need to optimize resource usage by implementing efficient algorithms, minimizing database queries, and utilizing caching mechanisms to enhance system responsiveness within resource constraints.
3. **Security Requirements:**
   * Constraints: The system must adhere to security best practices to protect user data and prevent unauthorized access.
   * Implications: Developers should prioritize implementing basic security measures such as secure authentication, data encryption, and input validation to mitigate common security risks within the project's constraints.
4. **Compliance Obligations:**
   * Constraints: The project may need to comply with relevant regulations such as GDPR or PCI DSS.
   * Implications: Developers must ensure basic compliance with applicable regulations by implementing necessary data protection measures and privacy policies within the project's limitations.
5. **Scalability Considerations:**
   * Constraints: The system should be designed to handle moderate increases in user traffic and data volume.
   * Implications: Developers should focus on designing a flexible and modular architecture that can accommodate moderate growth, utilizing scalable design patterns and efficient resource management techniques within the project's constraints.
6. **Budget and Time Constraints:**
   * Constraints: The project has limited resources and a tight timeline for development and deployment.
   * Implications: Developers need to prioritize essential features and functionalities that deliver the most value within the project's constraints, adopting an iterative development approach to incrementally deliver updates and improvements while managing time and budget effectively.

**Functional Requirements**

**| Use Case for User**:

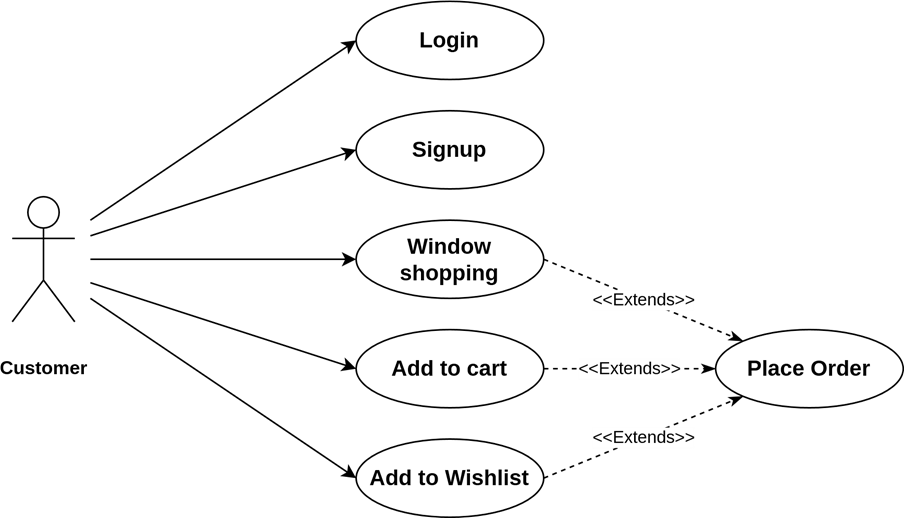


Fig. 1: Use Case For User

**| Use Case for Admin**:

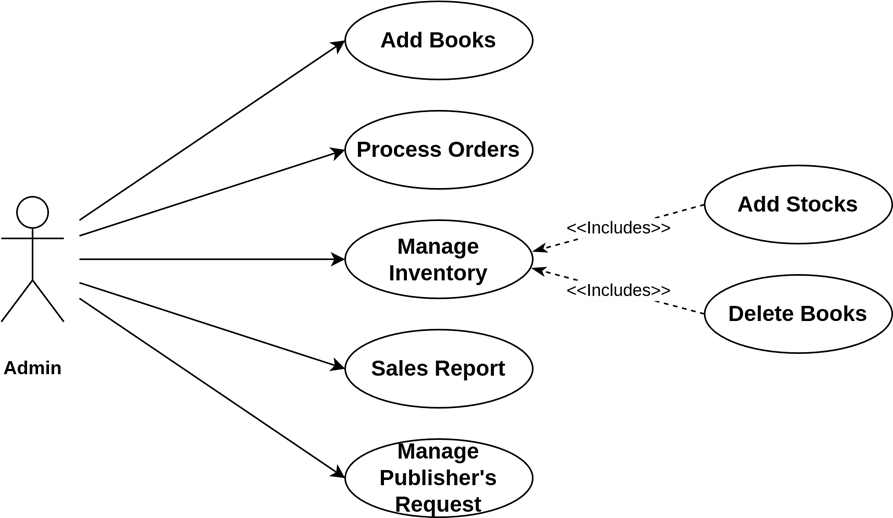


Fig. 2:User Case For Admin

**Non - Functional Requirements**

1. **Performance:**
   * The system should be responsive and performant, with fast loading times for web pages and quick response times for user interactions.
   * Response times for critical operations such as book search, order placement, and checkout should be within acceptable limits, even under peak load conditions.
   * The system should be able to handle concurrent user sessions without significant degradation in performance.
2. **Reliability:**
   * The system should be reliable and available, with minimal downtime and service interruptions.
   * It should have robust error handling and recovery mechanisms to gracefully handle errors and exceptions without crashing or losing user data.
   * Regular backups of the database should be performed to ensure data integrity and facilitate recovery in case of system failures.
3. **Security:**

* The system should adhere to industry-standard security practices to protect user data and prevent unauthorized access.
* User authentication and authorization mechanisms should be secure, with proper encryption of sensitive information such as passwords and payment details.
* The system should implement measures to protect against common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

1. **Scalability:**

* The system should be scalable, capable of handling increases in user traffic and data volume over time.
* It should be designed to scale horizontally by adding more server instances or resources as needed to accommodate growing demand.
* Scalability testing should be performed to ensure that the system can handle increased load without degradation in performance.

1. **Usability:**

* The system should be user-friendly and intuitive, with a clean and intuitive user interface that is easy to navigate.
* It should provide clear instructions and feedback to users to guide them through the various functionalities of the system.
* Accessibility features should be implemented to ensure that the system is usable by users with disabilities.

1. **Compatibility:**

* The system should be compatible with a wide range of devices and web browsers, ensuring a consistent user experience across different platforms.
* It should be responsive and adaptable to different screen sizes and resolutions, including mobile devices and tablets.
* Compatibility testing should be performed to verify that the system works correctly on various devices and browsers.

1. **Maintainability:**

* The system should be maintainable, with well-structured code and documentation that facilitate ongoing maintenance and updates.
* It should follow coding standards and best practices to ensure readability, modularity, and ease of debugging.
* Version control should be used to track changes to the codebase and facilitate collaboration among developers.

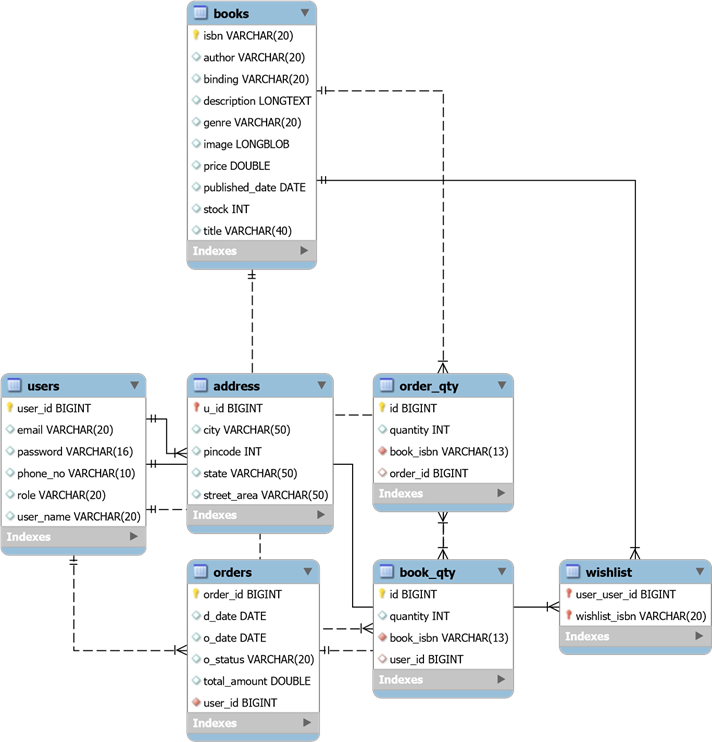


Fig. 3 : Complete Database

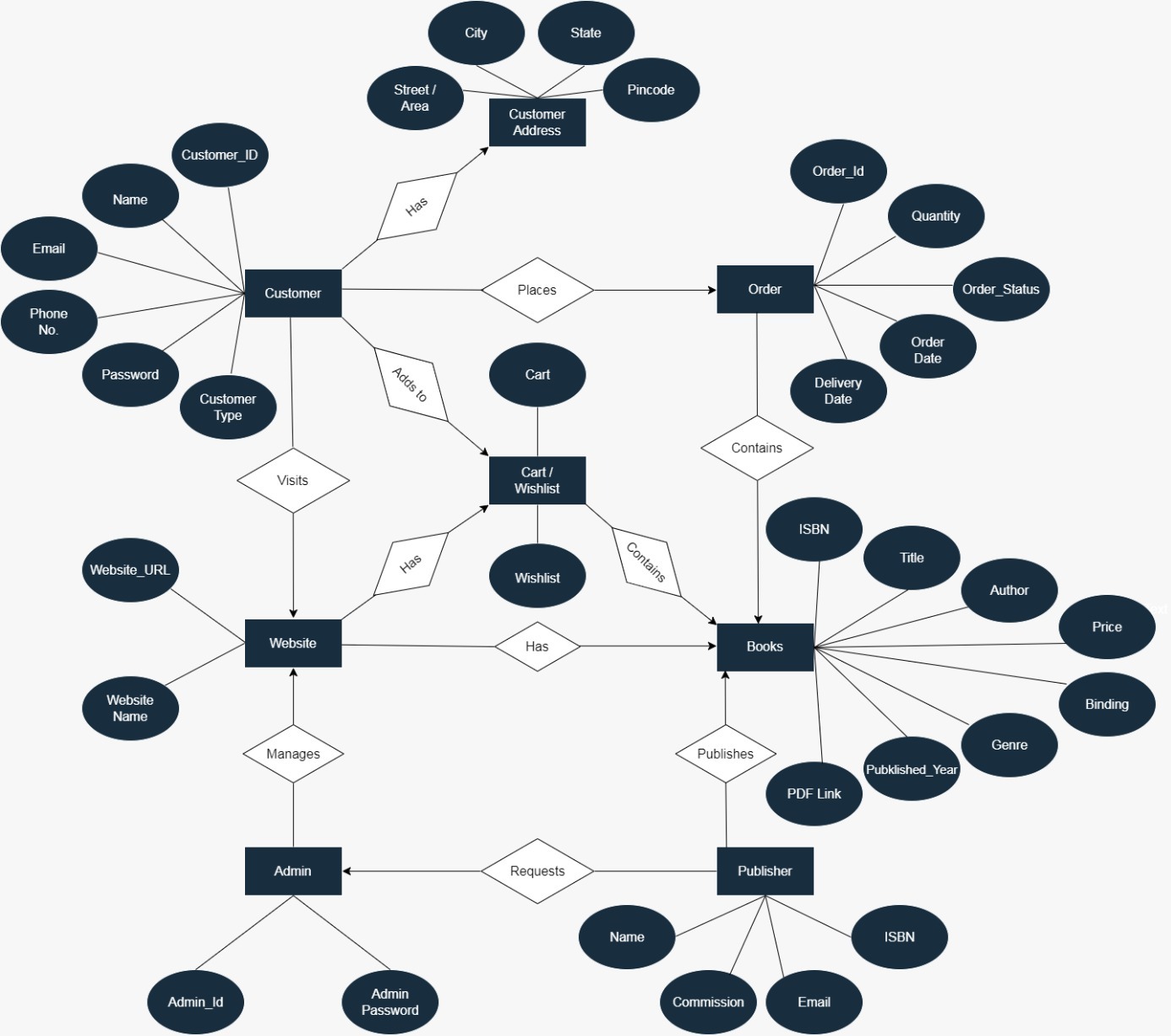


Fig. 4: ER Diagram

**Screen Shots**

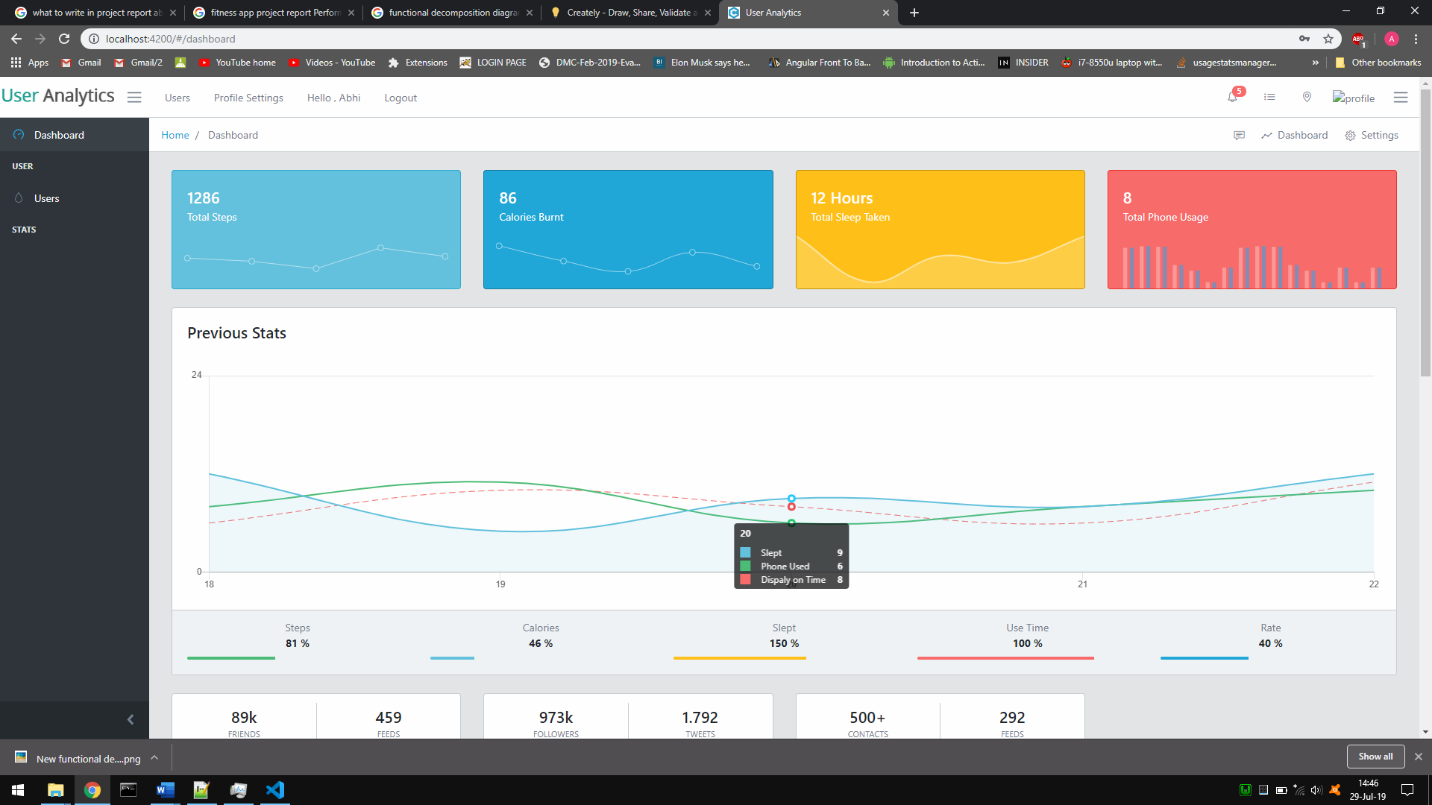


Fig. 4

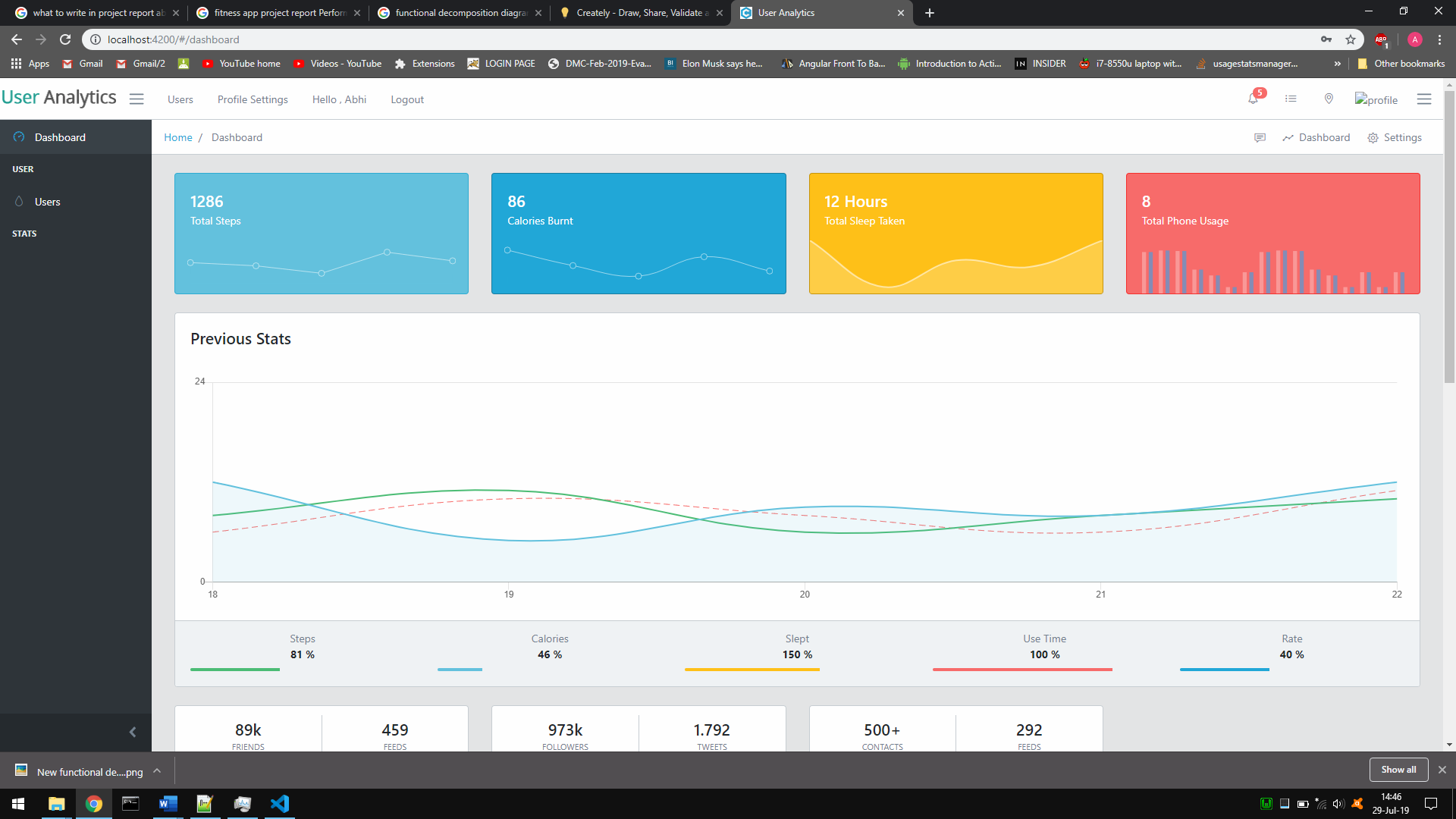


Fig. 5

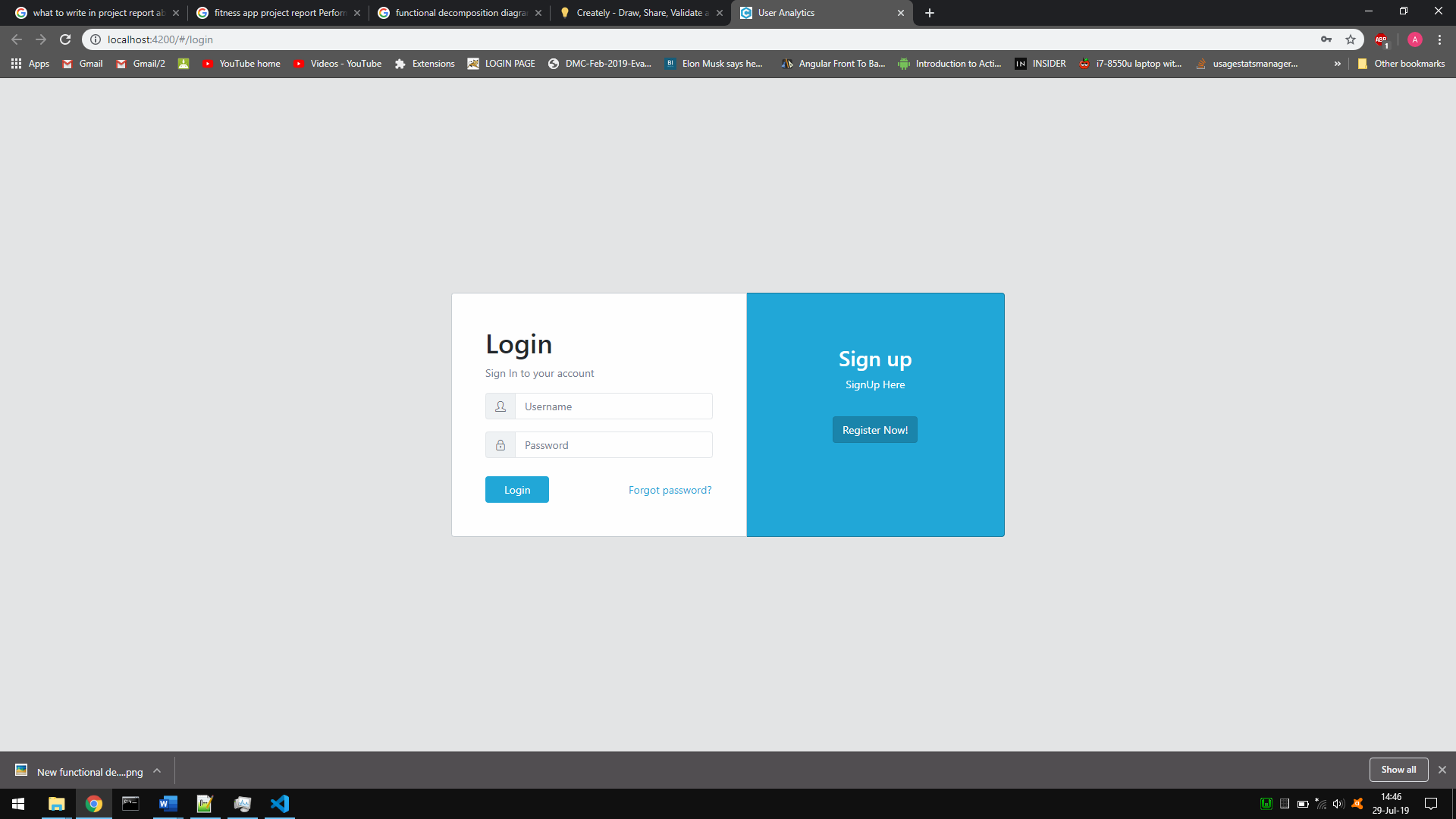


Fig. 6

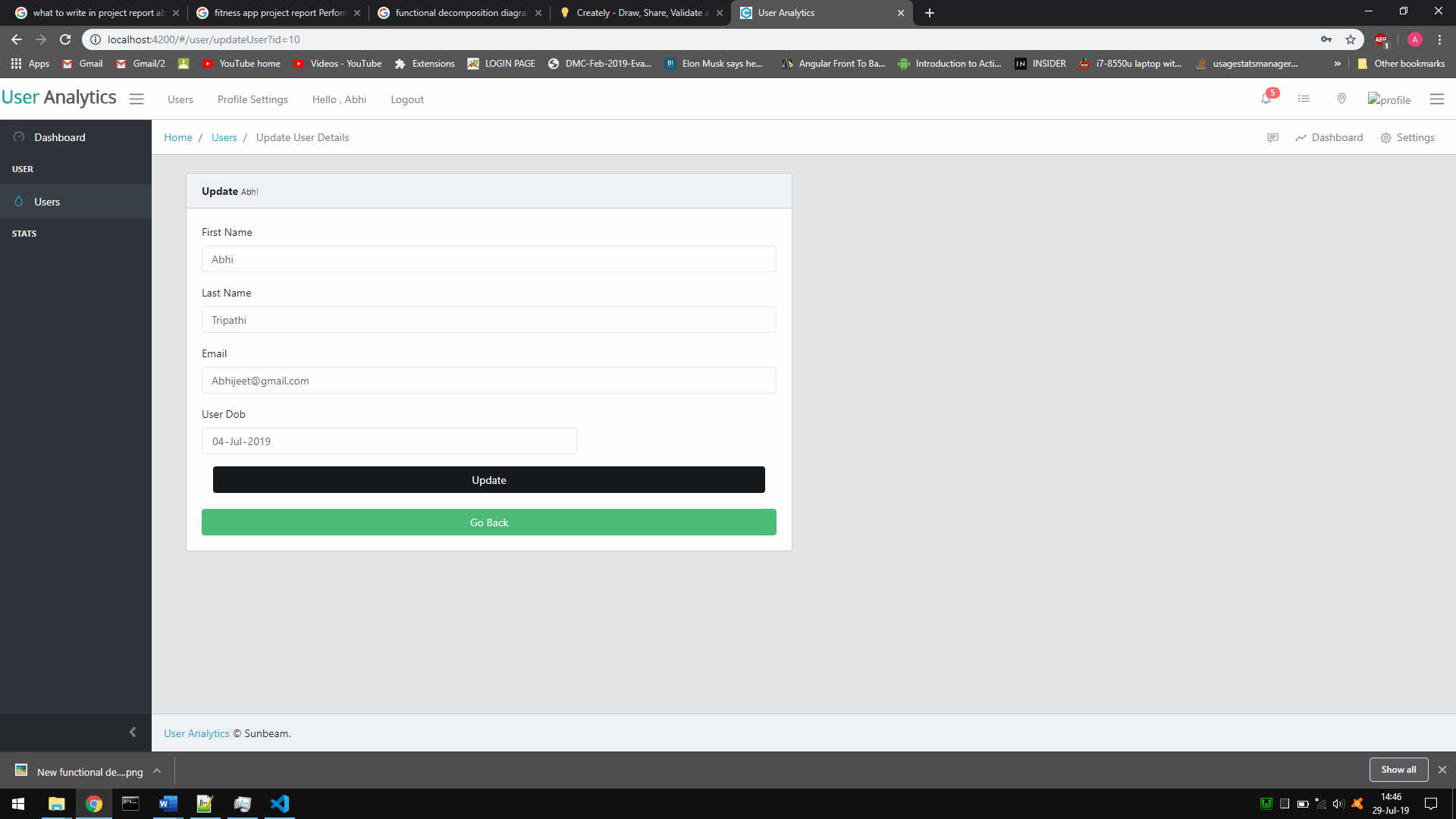


Fig. 7

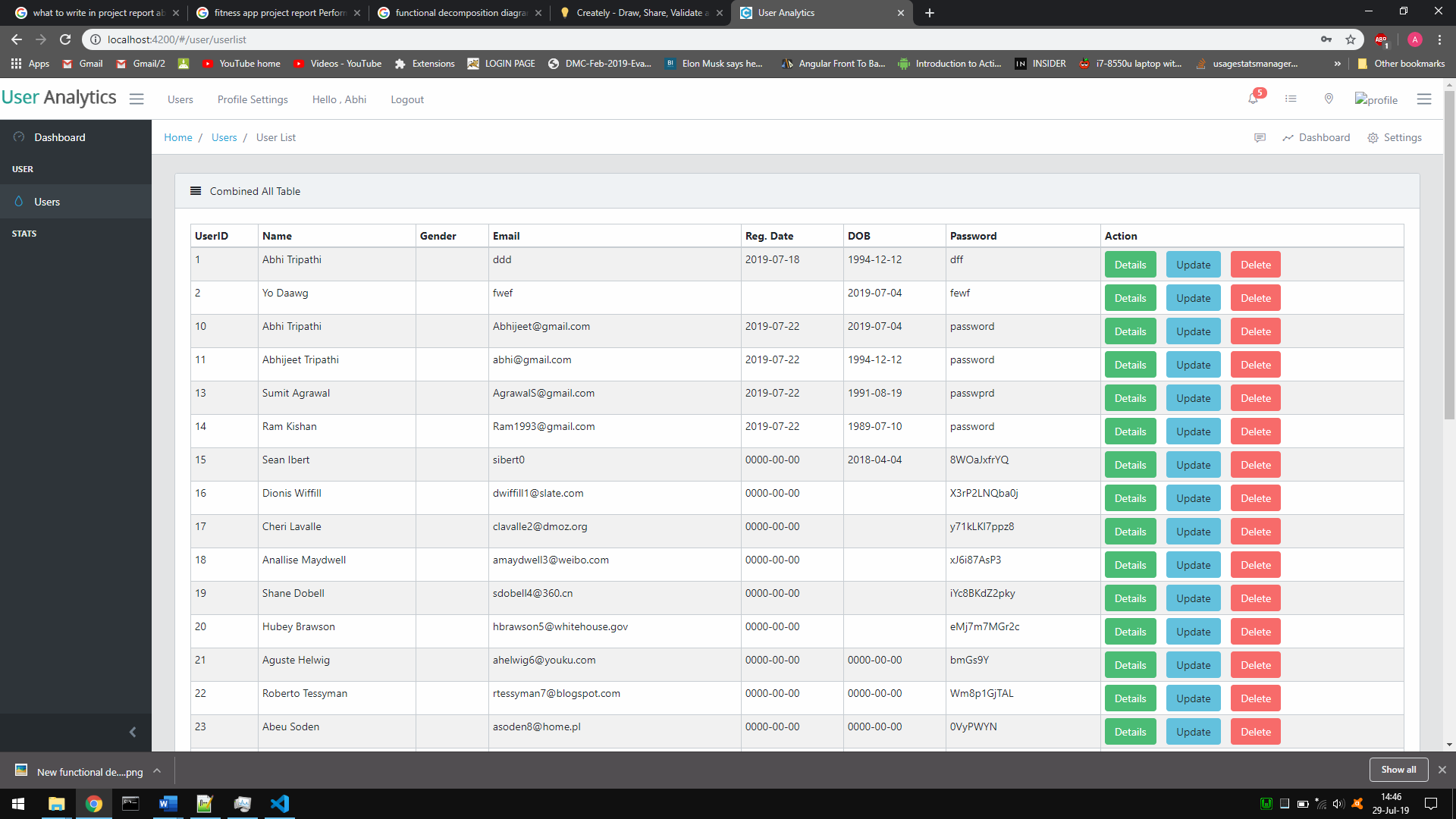


Fig. 8

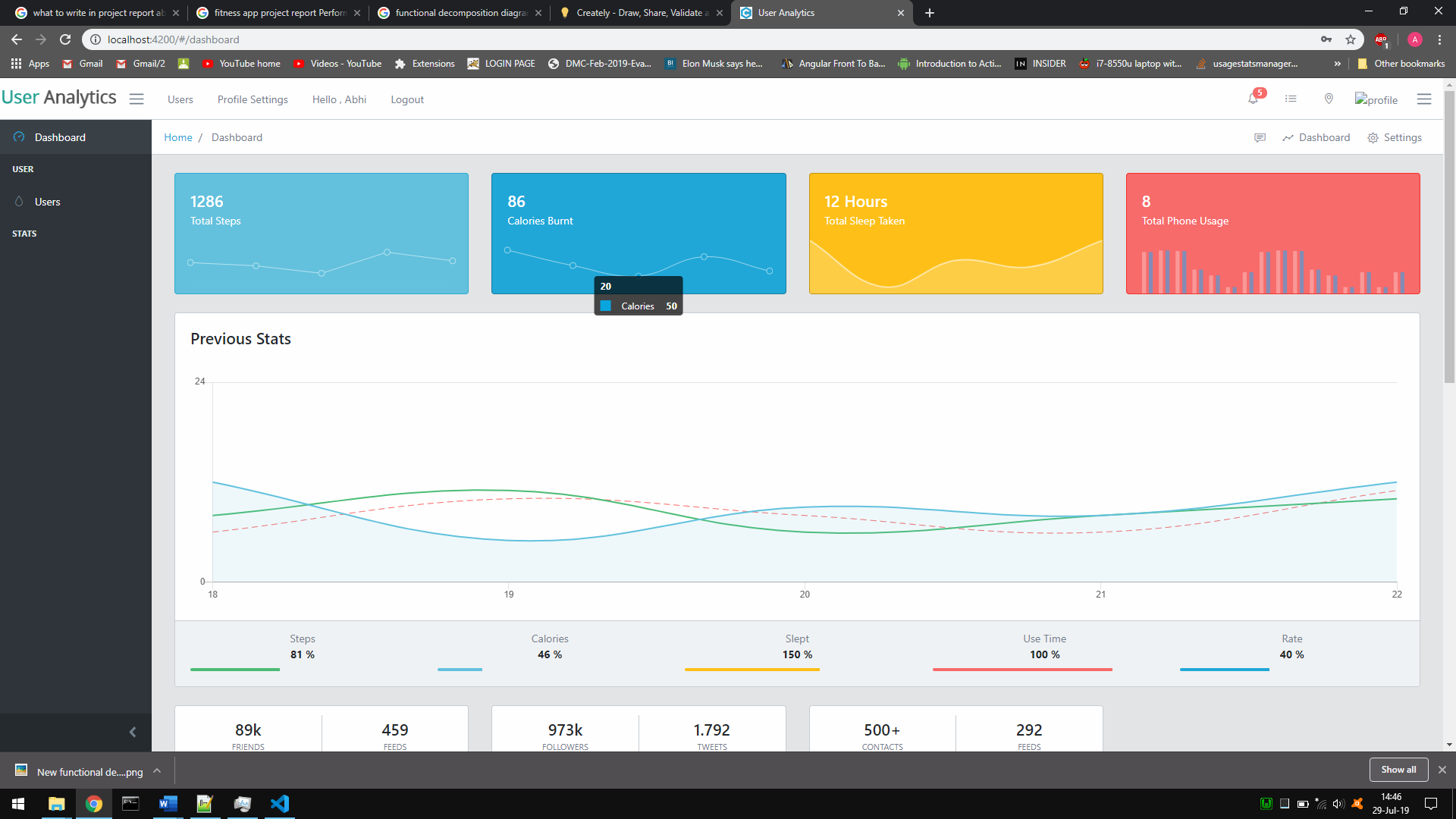


Fig. 9

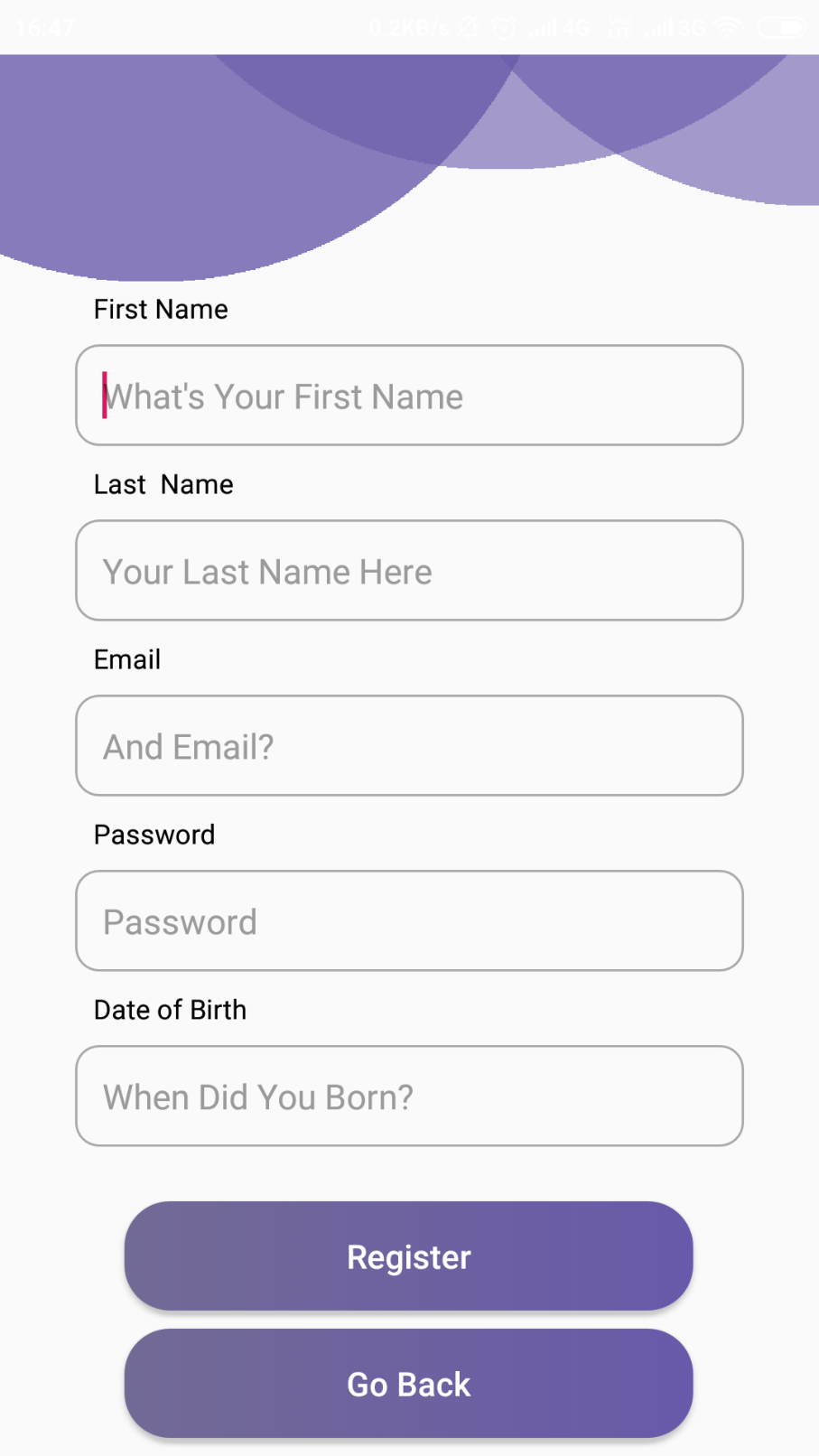


Fig. 10

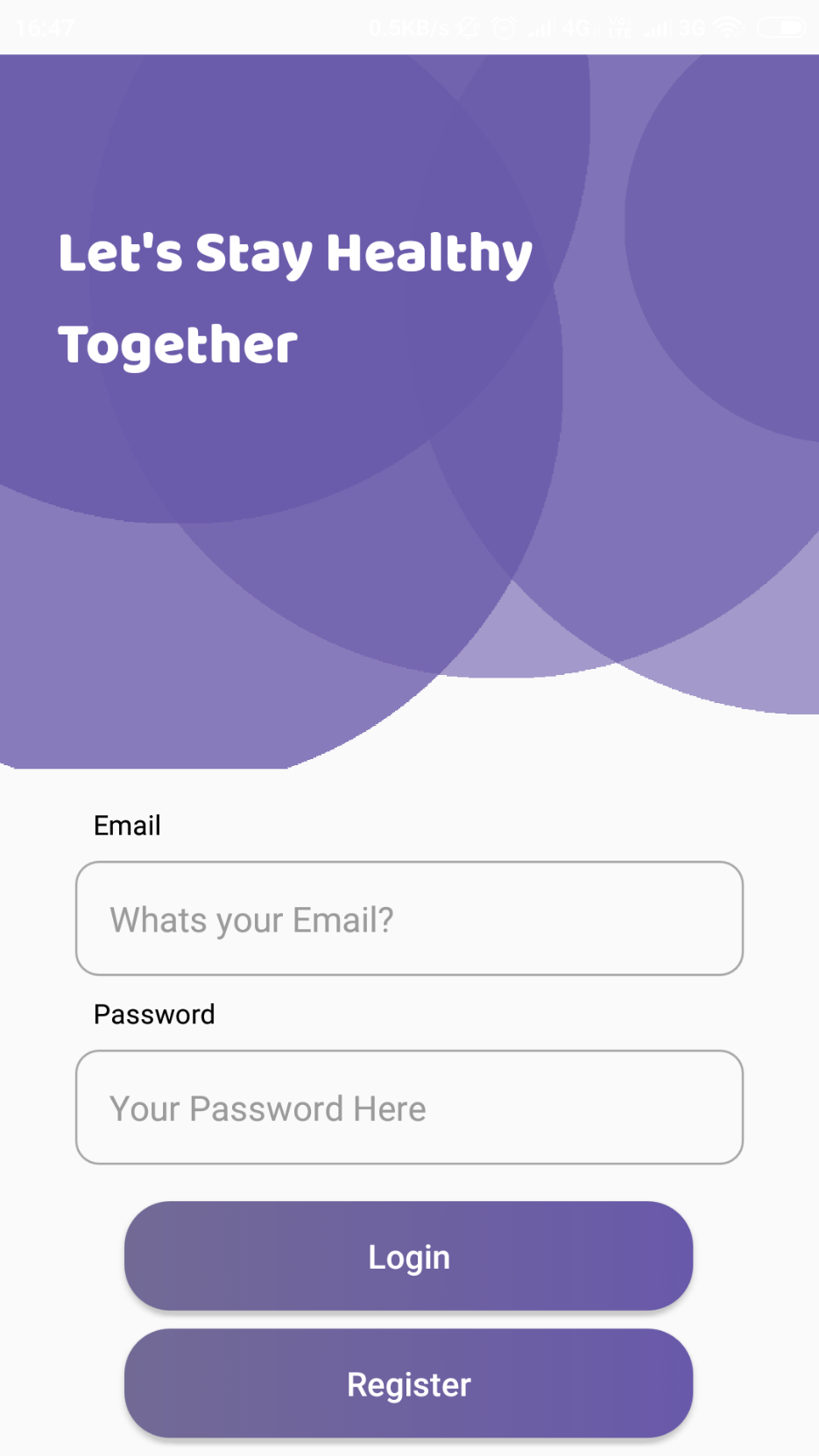


Fig. 11

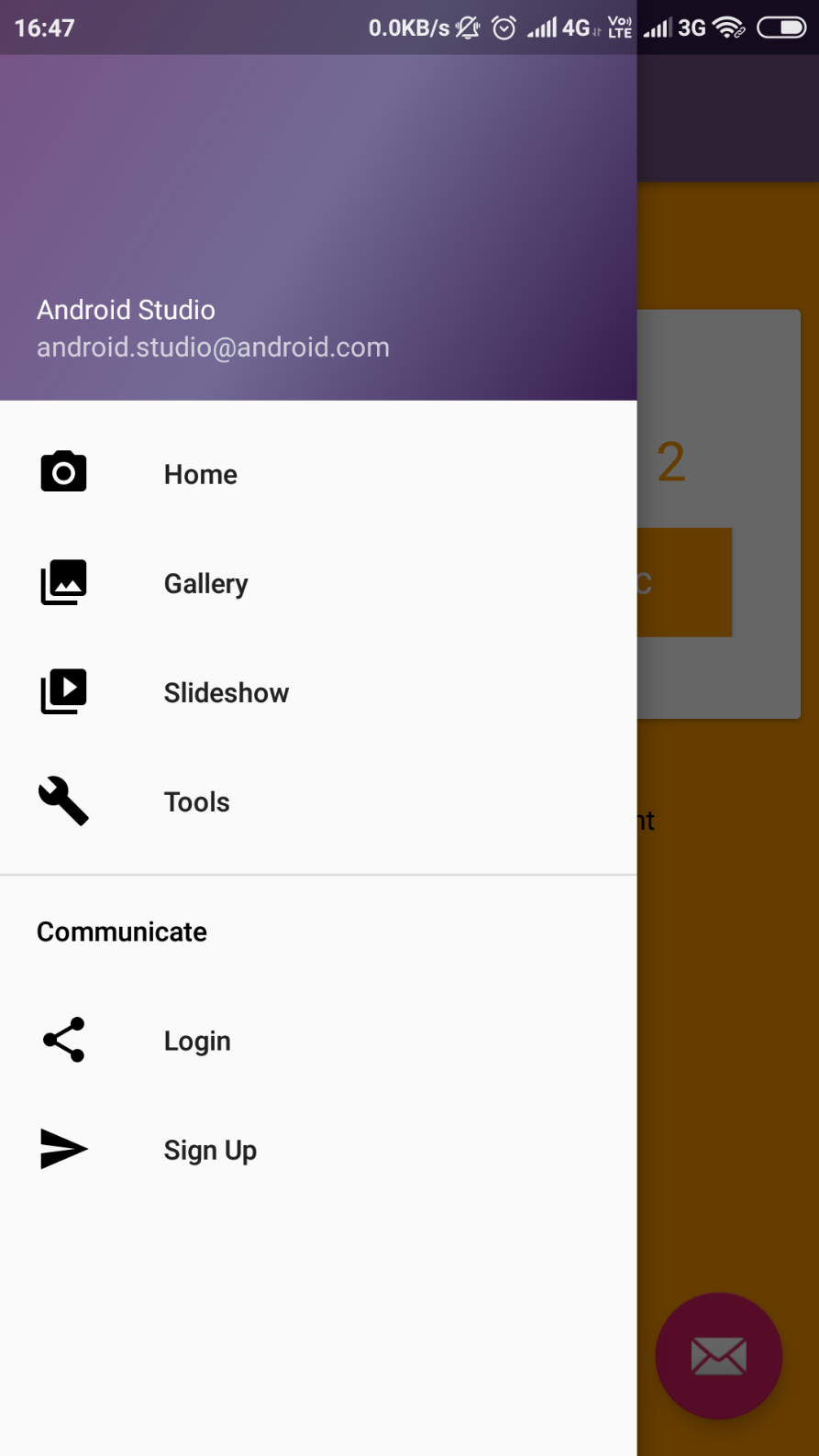


Fig. 12

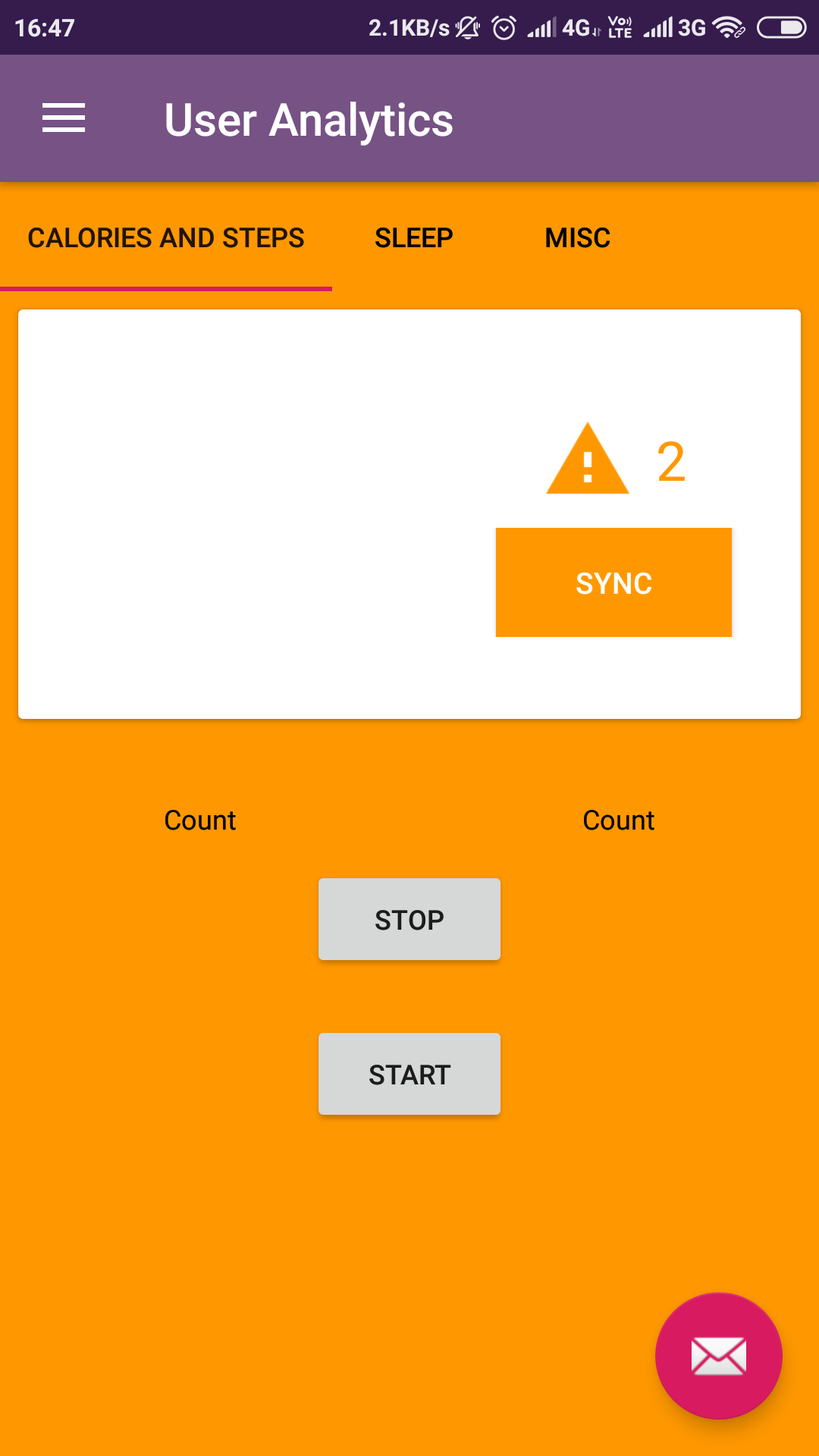


Fig. 13

**| Conclusion:**

The Online Book Store Management System, powered by Spring Boot and React, offers streamlined book sales and inventory control. With a focus on performance, reliability, and user experience, the system aims to enhance online bookstore operations. By adhering to security standards and scalability principles, it provides a robust platform for growth. Through its intuitive interface and efficient features, the system enables seamless browsing, purchasing, and management of books. Designed to meet the diverse needs of customers, administrators, and developers, it promises to be a valuable asset in the competitive e-commerce landscape.

**| Future Work :**

* **Sales Analysis for Admin:**

Implement sales analysis tools for administrators, allowing them to gain insights into sales trends, revenue performance, and inventory turnover. This feature will enable data-driven decision-making and help optimize business strategies.

* **Student Login:**

Introduce a student login feature, catering to a specific user class with tailored functionalities such as academic discounts, recommended reading lists, and integration with educational institutions' systems. This addition will expand the user base and enhance the platform's relevance in academic settings.

**| References:**

Time - Alarm Clock. Retrieved from https://play.google.com/store/apps/details?id=com.azumio.android.sleeptime&feature=search\_r esult#?t=W251bGwsMSwxLDEsImNvbS5henVtaW8uYW5kcm9pZC5zbGVlcHRpbWUiXQ Fitbit Inc. (n.d.). Fitbit. Retrieved from http://dev.fitbit.com/ Fragments. (n.d.). Retrieved March 23, 2015, from

http://developer.android.com/guide/components/fragments.html Gehring, J. (n.d.). Documentation. Retrieved March 27, 2015, from http://www.androidgraphview.org/documentation Google. (n.d.). E.

Automation and developer API - Sleep as Android. Retrieved from https://sites.google.com/site/sleepasandroid/doc/developer-api Lopresti, A. L., Hood, S. D., & Drummond, P. D. (2013). A review of lifestyle factors that contribute to important pathways associated with major depression: Diet, sleep, and exercise. Journal of Affective Disorders, 148, 12-27 Michaud, C. (n.d.). Pedometer Using Accelerometer Sensor. Retrieved March 26, 2015, from

http://nebomusic.net/androidlessons/Pedometer\_Project.pdf PELUSO MAM et al. Physical activity and mental health: the association between exercise and mood. CLINICS 60(1): 61-70, 2005. ProtoGeo. (n.d.). Moves. Retrieved from

https://play.google.com/store/apps/details?id=com.protogeo.moves&hl=en Recording Fitness Data. (n.d.). Retrieved March 23, 2015, from https://developers.google.com/fit/android/record Sleepiness:

Cognitive and Emotional Effects. (n.d.). Retrieved March 23, 2015, from http://www.webmd.com/sleep-disorders/excessive-sleepiness-10/emotions-cognitive?page=3 Sonstroem RJ, Morgan WP. Exercise and self-esteem rationale and model. Med Sci Sports Exerc 1989;21:329-37. ViewPager. (n.d.). Retrieved March 23, 2015, from

http://developer.android.com/reference/android/support/v4/view/ViewPager.html 40 Working with

the Fitness History. (n.d.). Retrieved March 23, 2015, from https://developers.google.com/fit/android/history Build software better, together. (n.d.). Retrieved

December 15, 2014, from https://github.com/ Tudor-Locke, C., & Bassett Jr, D. (n.d.). How Many Steps/Day Are Enough? Retrieved November 15, 2014, from

http://www.health.utah.edu/peak/docs/Tudor Locke Paper.pdf National Sleep Foundation

Recommends New Sleep Times. (n.d.). Retrieved February 27, 2015, from http://sleepfoundation.org/media-center/press-release/national-sleep-foundationrecommends-new-sleep-times