**Major Project**

**Project - Synopsis**

MCA - III Sem

Subject Code: CA7131

Submitted By

Kuldeep singh

23FS20MCA00076

Faculty Coordinator

Dr. Linesh Raja, Associate Professor

**applications. It encompasses a variety of tasks, including web design, coding, content creation, and server configuration. Essentially, web development transforms ideas into functional online experiences.**

**There are two main components of web development:**

**1. \*\*Front-End Development: This involves everything that users interact with directly in their web browsers. Front-end developers use languages like HTML, CSS, and JavaScript to create visually appealing and user-friendly interfaces**

**2. \*\*Back-End Development\*\*: This focuses on the server-side of applications, handling data storage, server configuration, and application logic. Back-end developers typically work with languages such as PHP, Python, Ruby, or Java and manage databases to ensure everything runs smoothly behind the scenes.**

**In today’s digital age, effective web development is crucial for businesses and individuals alike, enabling them to establish an online presence, engage with users, and drive growth. Whether it’s a simple blog or a complex e-commerce platform, web development is at the heart of the online experience.**

**KULDEEP SINGH**

**23FS20MCA00076**

**PHONE NUMBER: 9116418169**

**EMAIL: HARSHUBNA@GMAIL.COM**

**SKILLS: REACTJS NODEJS EXPRESS MONGODB GITBASH GITHUB**

**BACKEND: NODEJS EXPRESS POSTGRESQL**

**OBJECTIVES**

**1. facilitate Secure Transactions: Ensure that all financial transactions are processed safely and securely, protecting users' personal and financial information.**

**2. User-Friendly Interface: Create an intuitive and easy-to-navigate platform that enhances the user experience, making transactions straightforward for all users.**

**3. Speed and Efficiency: Optimize transaction processing times to ensure quick and seamless payments, minimizing user wait times.**

**4. Wide Range of Payment Options: Offer multiple payment methods (credit/debit cards, digital wallets, bank transfers) to cater to diverse user preferences.**

**5. Financial Inclusion: Provide access to financial services for underserved populations, enabling them to participate in the digital economy.**

**6. Robust Customer Support: Establish a responsive customer service system to assist users with inquiries and resolve issues promptly.**

**7. Educational Resources: Offer guides and resources to help users understand online transactions, budgeting, and financial management.**

**8. Promote Trust and Transparency: Foster user trust by clearly communicating fees, terms, and conditions associated with transactions.**

**9. Adapt to Market Trends: Continuously update features and services in response to changing market demands and technological advancements.**

**10. Data Analytics and Insights: Utilize data analytics to monitor user behavior, improve services, and tailor offerings based on user needs and preferences.**

**11. Continuous Innovation**: Stay ahead of market trends by regularly updating features and introducing new technologies that enhance the overall user experience.

**12.Partnership Development**: Foster strategic partnerships with merchants, businesses, and financial institutions to broaden service offerings and reach more users.

**13. Loyalty and Rewards Programs**: Implement programs that reward users for transactions, encouraging engagement and increasing customer retention.

**14. Market Expansion**: Grow the user base by entering new markets and demographics, making digital payments accessible to a wider audience.

DFD (DATA FLOW DIAGRAM)

Level 0 DFD (Context Diagram)

Entities

- UserIndividuals or businesses using the platform for transactions.

- BankFinancial institutions involved in processing transactions.

- MerchantBusinesses that receive payments via PayBuddy.

Processes

- PayBuddy SystemCentral process that manages all transactions and user interactions.

Data Flows

- Users submit payment requests and receive confirmations.

- Payment information is sent to and from banks.

- Merchants receive payment notifications

Level 1 DFD

1. Process: User Registration

InputUser details (name, email, phone, etc.)

OutputUser account created confirmation.

2. Process: Payment Processing

InputPayment request (amount, recipient, payment method).

OutputTransaction confirmation, payment status.

3. Process: Transaction History

Input: User request for transaction history.

Output: Display of past transactions.

4. Process: Account Management

Input: User updates (profile, payment methods).

Output: Confirmation of updated information.

5.Process: Merchant Payment Notification

Input: Payment processed data.

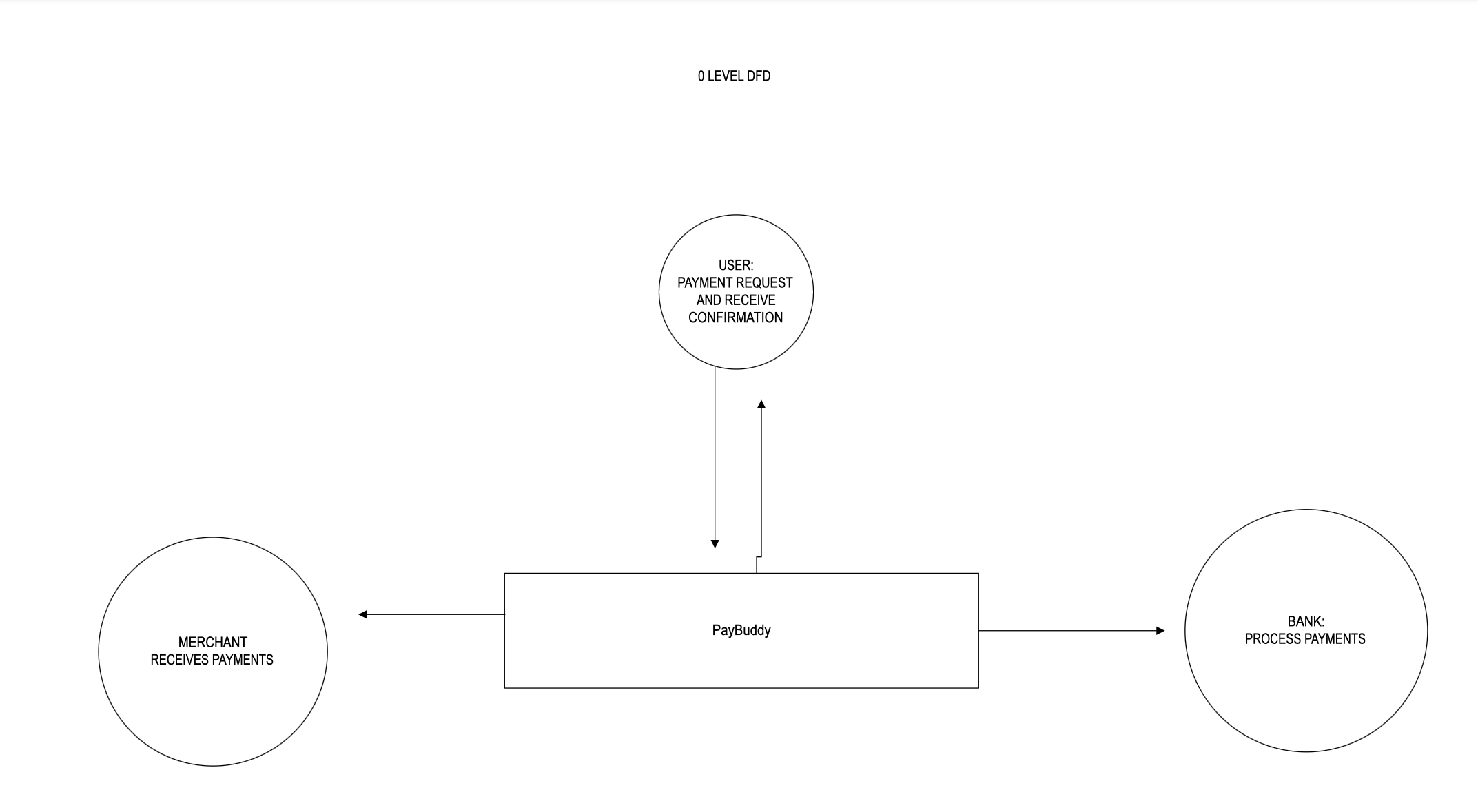
Output: Payment received confirmation to the merchant.

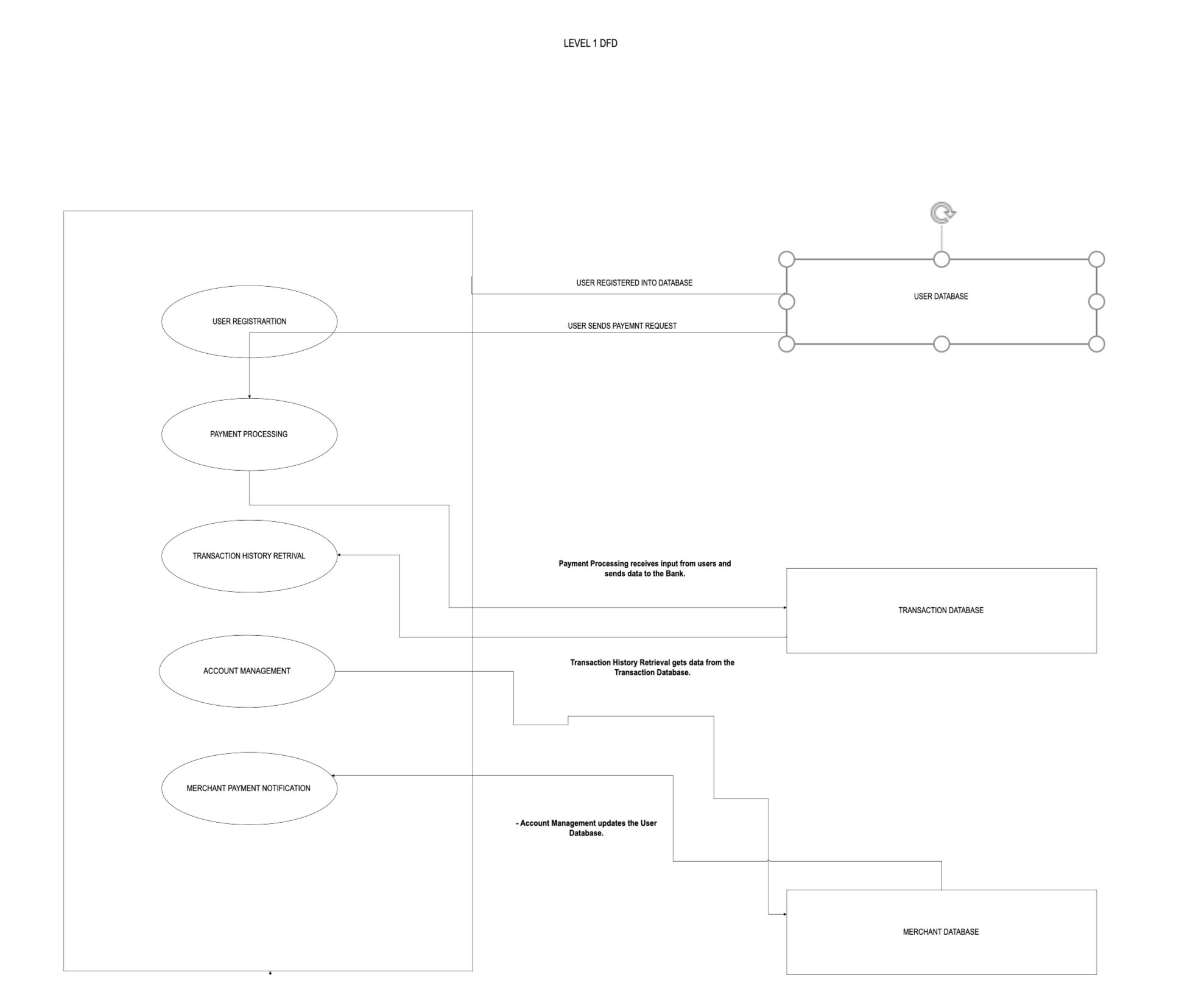
Data Stores:

User Database: Stores user profiles, authentication details, and preferences.

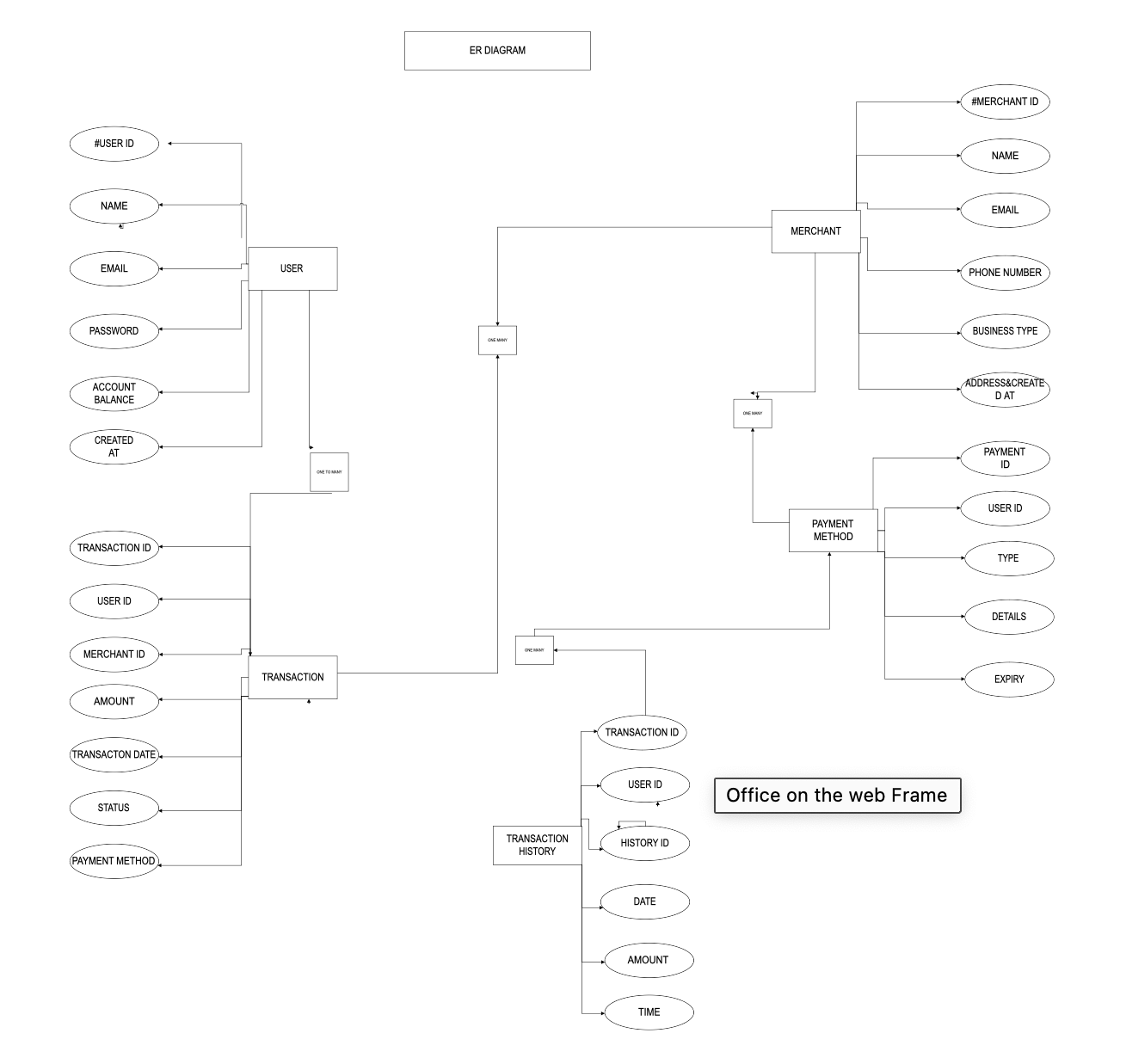
Transaction Database: Records all transactions processed through PayBuddy, including amounts and dates.

Merchant Database: Contains information about registered merchants and their transaction histories.





ER DIAGRAM



TOOLS, PLATFORM TECH AND SYSTEM

HARDWARE AND

SOFTWARE REQUIREMENTS

1. Tools and Platform

- Development Frameworks:

- Frontend: React, Angular for building responsive user interfaces.

- Backend:Node.js, ASP.NET for server-side logic, javascript.

- Database Management:

- Relational Databases : MySQL, PostgreSQL for structured data storage no use of my sql in this project we will use mongodb.

- NoSQL Databases important:MongoDB or Firebase for flexible data models.

- Payment Gateway Integration:

- Stripe, PayPal, or Braintree for handling payment processing securely and other free resources

- Version Control:

- Git with platforms like GitHub or GitLab for source code management and collaboration.

- Deployment Platforms:

- AWS, Azure, or Heroku for hosting the application and managing infrastructure.

- Design Tools:

- Figma UI/UX design and prototyping.

2. Hardware Requirements

- Server Specifications: hosting and other property

- CPU: Minimum 4-core processor (e.g., Intel i5 or equivalent).

- RAM: At least 8 GB for basic operations; 16 GB or more for higher traffic.

- Storage: SSD with a minimum of 100 GB, scalable based on user data and traffic.

- Network: High-speed internet connection with low latency.

- Development Machine:

- CPU: Minimum dual-core processor.

- RAM: At least 8 GB for smooth development.

- Storage: SSD preferred, with at least 256 GB for development tools and software.

3. Software Requirements

- Operating System:

- Windows, macOS, or Linux (Ubuntu preferred for server environments).

- Development Software:

- Text Editors/IDEs: Visual Studio Code, IntelliJ IDEA, or Sublime Text.

- Web Browsers:Chrome, Firefox, or Edge for testing.

- Database Management Tools: MySQL Workbench, pgAdmin for managing databases.

- Frameworks and Libraries:

- Frontend: Libraries such as Bootstrap or Tailwind CSS for responsive design.

- Backend:Express.js (for Node.js) or Flask.

- Security Software: extra functions

- SSL certificates for secure data transmission.

- Firewalls and anti-virus software for server protection.

- Collaboration Tools;

- Slack or Microsoft Teams for team communication.

- Trello or Jira for project management and task tracking.

4. Additional Considerations

- Backup Solutions:Regular automated backups of databases and application files.

- Monitoring Tools:Tools like Google Analytics for user tracking, and New Relic or Datadog for performance monitoring.

- ComplianceEnsure :compliance with relevant regulations (e.g., PCI-DSS for payment processing).

REFERENCES

1. W3Schools: [W3Schools](https://www.w3schools.com/

2. freeCodeCamp: [freeCodeCamp](https://www.freecodecamp.org/)

- Interactive coding lessons on web development, including projects and certifications.

Frameworks and Libraries

3. React: [React Documentation](https://reactjs.org/docs/getting-started.html)

4. Angular: [Angular Documentation](https://angular.io/docs)

Backend Development

5. Node.js: [Node.js Documentation](https://nodejs.org/en/docs/)

- Official documentation for building server-side applications using Node.js.

6. Django: [Django Documentation](https://www.djangoproject.com/)

- Full documentation for the Django web framework in Python.

UI/UX Design

7. Figma: [Figma Resources](https://www.figma.com/resources/)

- Tools and resources for designing user interfaces and prototypes.

Version Control

8. GiT: [Pro Git Book](https://git-scm.com/book/en/v2)

- Free book on Git, covering everything from basics to advanced topics.

9. GitHub: [GitHub Guides](https://guides.github.com/)

- Guides on how to use GitHub for version control and collaboration.

10. AWS: [AWS Documentation](https://docs.aws.amazon.com/)

- Comprehensive resources for using AWS services for hosting and managing applications.

11PayPal: [PayPal Developer](https://developer.paypal.com/docs/)

- Guides and APIs for integrating PayPal payments.

Additional Learning Resources

12. Coursera: [Web Development Courses](https://www.coursera.org/browse/computer-science/mobile-and-web-development)

- Online courses on various web development topics.

**THANK YOU.**