INTRODUCTION

- **SonicFund** is designed to transform learning with an innovative online platform offering:
  - High-quality educational resources.
  - Personalized learning experiences tailored to individual needs.
- The platform addresses the challenges of traditional education by providing:
  - Flexible and accessible learning solutions.
  - Interactive tools to meet the needs of diverse learners.
- **Key Features:**
  - Combines advanced technologies and modern teaching methods to make education accessible for everyone.
  - Promotes collaboration and knowledge sharing between learners, educators, and experts.
  - Fosters a vibrant community to spark curiosity and encourage intellectual growth.
- With features like curated content, personalized learning pathways, and interactive tools, SonicFund ensures:
  - An engaging and user-friendly experience.
  - Improved learning retention and academic success.
  - Positive social impact by transforming education for people from all backgrounds.

TECHNOLOGY USED

- **MongoDB** serves as the flexible and scalable database solution, accommodating the diverse needs of a dynamic educational platform.

- **Express.js** simplifies server-side development, streamlining the creation of robust backend systems to handle user interactions and data management.

- **React.js** powers the dynamic and interactive user interface, offering a rich learning experience with its component-based architecture.

- **Node.js** enables high-performance server-side execution, facilitating real-time interactions and fast data processing for an optimal user experience.

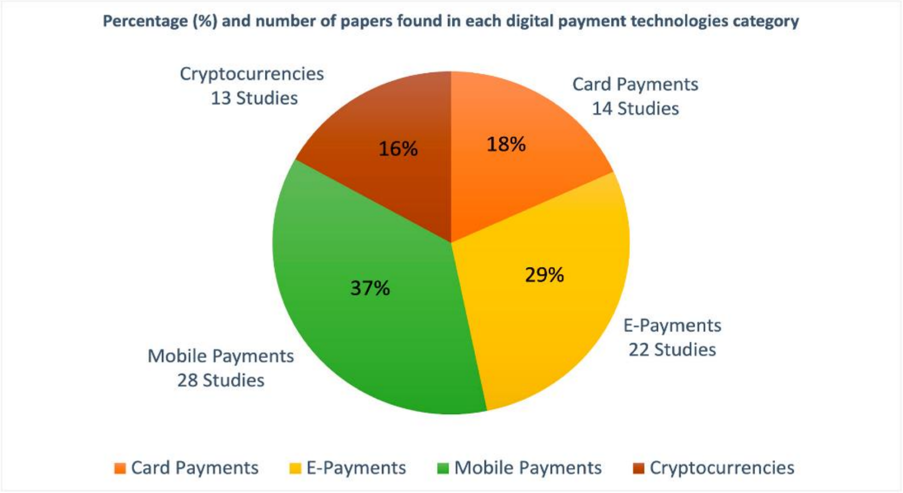
OBJECTIVES

- **Facilitate Access to Financial Services**
  - Make financial transactions accessible to users worldwide.
  - Offer secure and efficient financial tools regardless of:
    - Geographical location.
    - Socioeconomic status.
    - Financial background.
- **Personalize Transaction Experiences**
  - Tailor financial services to each user's unique needs and preferences.
  - Use advanced algorithms to:
    - Provide customized financial recommendations.
    - Track transaction histories.
    - Offer targeted support for achieving financial goals.
- **Foster Collaboration and Engagement**
  - Build a vibrant financial community with interactive features:
    - Forums for discussion.
    - Community-driven advice.
    - Tools for collaboration among users, experts, and businesses.
  - Enable users to share insights, exchange ideas, and learn from others.
- **Promote Financial Literacy**
  - Provide diverse resources for users of all ages and backgrounds.
  - Empower individuals to:
    - Confidently manage their finances.
    - Adapt to changing financial needs.
    - Achieve long-term financial goals.
- **Drive Innovation in Financial Services**
  - Continuously improve the platform by:
    - Adopting the latest technologies and industry best practices.
    - Incorporating user feedback.

RELATED STUDIES

- **Existing Digital Wallets**: Paytm, Google Pay, PhonePe, and similar payment platforms are widely used.
- **Issues in Current Solutions**:
  - **Security Gaps**: Weak authentication, inadequate encryption, and vulnerabilities to phishing.
  - **Limited User Control**: Users have little control over their data and payment flow.
  - **Research Insight**: Users prefer apps that offer strong security, privacy, multi-platform support, and fast payments. This insight forms the foundation for **SonicFund's development**.
  - **Privacy Flaws**: Platforms collect and track user data for marketing purposes.

Numbers of studies related to each payment technologies



FEASIBILITY STUDY

1. Technical Feasibility

- **Resource Availability:**
    - Ensure access to essential resources like hardware, software, and skilled personnel for development and maintenance.
- **Technology Stack:**
    - Evaluate the MERN stack for:
        - Scalability and security.
        - Compatibility with financial systems.
- **Infrastructure Requirements:**
    - Identify the needed infrastructure, including:
        - Web servers and secure payment gateways.
        - Databases and sufficient network bandwidth.

2. Economic Feasibility

- **Cost Analysis:**
    - Assess all costs involved:
        - Development and hosting.
        - Maintenance and potential revenue sources (e.g., transaction fees, premium features, partnerships).

3. Operational Feasibility

- **Organizational Capabilities:**
    - Review organizational strengths in:
        - Project management and technical expertise.
        - Operational readiness to develop and maintain the platform.
- **Performance Optimization:**
    - Continuously improve platform performance by:
        - Streamlining payment processing.
        - Enhancing API endpoints and frontend responsiveness.
- **Training and Support:**
    - Implement effective onboarding programs for:
        - Staff, partners, and users.

METHODOLOGY

Agile Methodology Steps for SonicFund

1. Plan

- Define project objectives, scope, and requirements based on:
  - Stakeholder feedback.
  - Project guidelines.
- Break down the project into tasks such as:
  - Market research.
  - Designing transaction workflows.
  - Coding assignments.
- Set timelines and milestones aligned with the financial roadmap.

2. Design

- Visualize the platform structure with:
  - Wireframes, flowcharts, or diagrams.
- Create prototypes using tools like Sketch or Figma, incorporating stakeholder feedback.
- Ensure designs prioritize:
  - User needs.
  - Security and usability standards.

3. Develop

- Begin coding tasks based on the design specifications.
- Use version control systems (e.g., Git) for collaboration and managing code changes.
- Break down development into smaller modules such as:
  - Payment gateways.
  - User authentication.
  - Transaction dashboards.
- Track progress for each module.

4. Test

- Conduct:
  - **Unit testing** for individual components.
  - **Integration testing** to ensure seamless module interactions (e.g., payments, accounts, notifications).

- Perform code reviews and demo sessions to gather feedback.

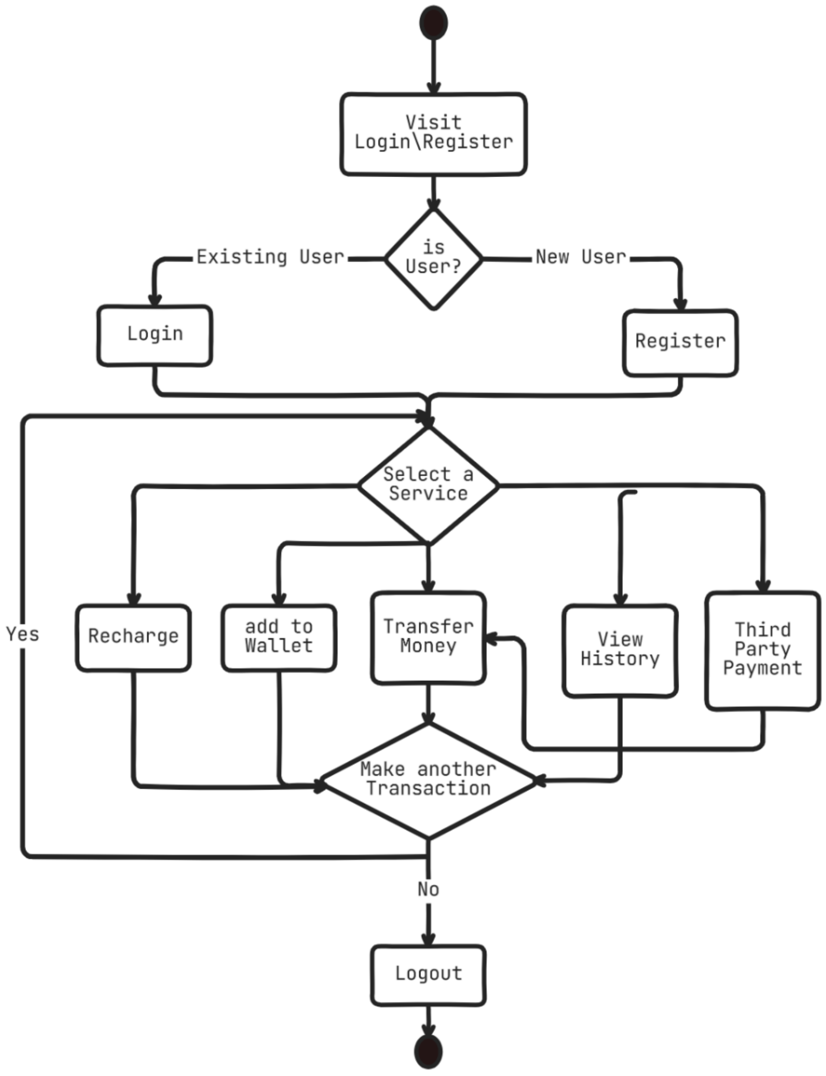
5. Deploy

- Prepare for platform launch by:
  - Ensuring code is documented, secure, and formatted.
  - Packaging project files and dependencies for deployment.
- Double-check deployment requirements, such as:
  - Security certifications.

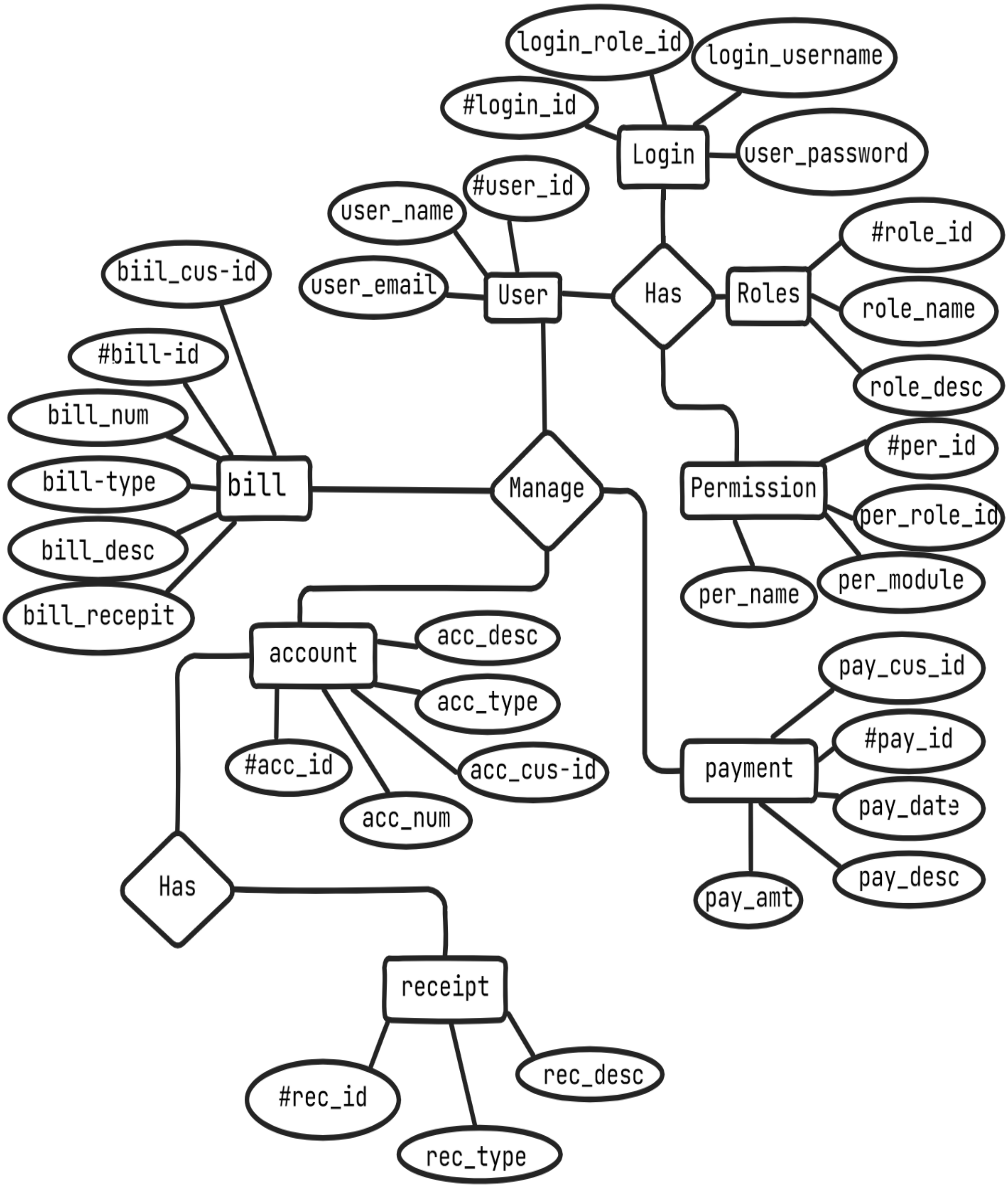
6. Review

- Reflect on project outcomes by:
  - Identifying challenges faced and lessons learned.
  - Collecting feedback via platform demos and user testing.
- Document insights for future iterations and continuous improvement.

FLOW DIAGRAM



E- R DIAGRAM


login_role_id
login_username
#login_id
Login
user_password
#user_id
user_name
user_email
User
Has
Roles
#role_id
role_name
role_desc
biil_cus-id
#bill-id
bill_num
bill-type
bill
bill_desc
bill_recepit
Manage
Permission
#per_id
per_role_id
per_module
per_name
account
acc_desc
acc_type
#acc_id
acc_cus-id
acc_num
pay_cus_id
#pay_id
payment
pay_date
pay_desc
pay_amt
Has
receipt
#rec_id
rec_desc
rec_type

SOFTWARE AND HARDWARE REQUIREMENTS

a) Software Requirements:

1. Operating System: Windows 10/11 or Linux distribution (e.g., Ubuntu)

2. Development tools:

- Text editor or IDEs like VS-CODE, Notepad
- Web Browser: Google chrome, Brave
- Command line interface: Terminal (Linux), Command prompt

3. Version Control system: Git, Git Client (e.g. GitHub Desktop)

4. Database Management System: MongoDB, MongoDB Compass

5. Development Framework and libraries:

MERN Stack:

- MongoDB: NoSQL Database
- Express.js: Web application framework
- React.js: JavaScript library for building user interfaces
- Node.js: JavaScript runtime environment

Additional libraries and frameworks as needed:

- Tailwind-CSS, Material-UI, or similar for frontend UI components
- Axios or similar for making HTTP requests
- Redux or similar for state management in React.js applications

b) Hardware requirements:

- Desktop or Laptop computer with adequate processing power and memory.
- **Minimum requirements:**
- Intel Core i5 processor or equivalent
- 8GB RAM
- 250GB SSD or 1TB HDD storage

Recommended:

- External monitor(s) for multi-tasking and improved productivity (optional)
- Mouse and keyboard for ergonomic use (optional)

Expected Outcome(s)

- **Enhanced Security**: Multi-factor authentication and encryption for safer transactions.
- **Better User Experience**: Intuitive, easy-to-use interface.
- **Faster Payments**: Quick and seamless transaction processing.
- **Privacy**: Minimal data collection, ensuring user privacy.
- **Multiple Payment Methods**: Supports UPI, cards, and net banking.
- **Scalability**: Can handle growing transaction volumes.

REFERENCES

➢ **Reference Books**

- "MERN Quick Start Guide: Build web applications with MongoDB, Express.js, React, and Node" by Eddy Wilson Iriarte Koroliova
- "Node.js Web Development: Server-side web development made easy with Node 14 using practical examples" by David Herron

➢ **Reference Websites**

- What is Agile methodology? (A beginner’s guide) by Sarah Loyan

  https://asana.com/resources/agile-methodology
- MongoDB Documentation

  https://docs.mongodb.com/
- Express.js Documentation

  https://expressjs.com/en/resources/glossary.html
- React.js Documentation

  https://legacy.reactjs.org/docs/
- Node.js Documentation

  https://nodejs.org/docs/
- Inspired by

  https://100xdevs.com/