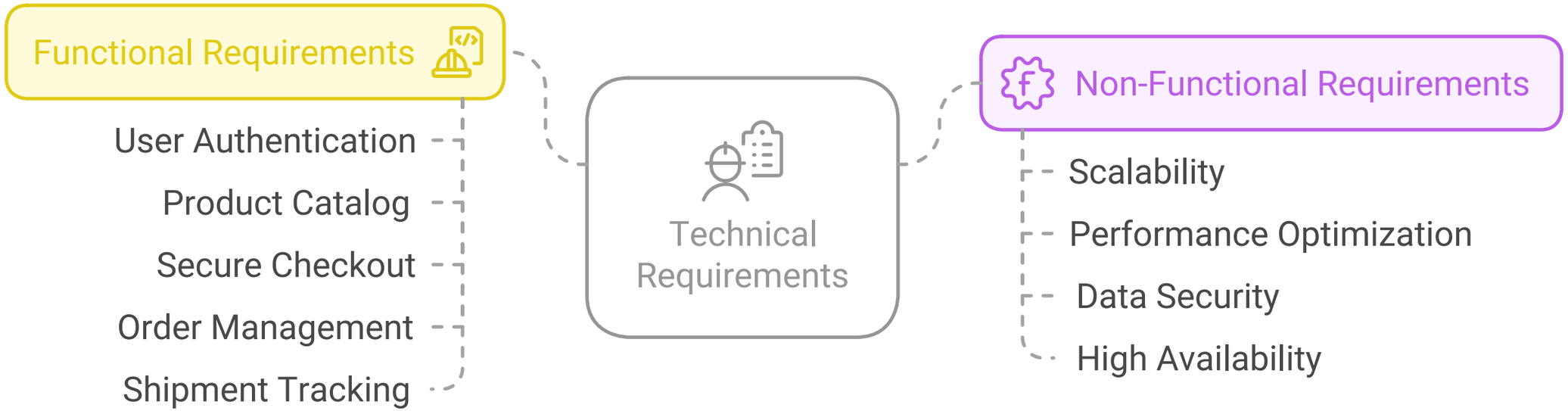
Day 2: Technical Planning for Premii Woods

1. Define Technical Requirements
    - **Functional Requirements:**
        - User authentication and authorization (login/sign-up via email or social accounts).
        - Product catalog with search and filtering.
        - Secure checkout process with payment integration (e.g., QuickPay, JazCash).
        - Order management (cart, checkout, and order history).
        - Real-time shipment tracking.
    - **Non-Functional Requirements:**
        - Scalability to handle growing traffic.
        - Performance optimization for fast page loading.
        - Data security for user and payment information.
        - High availability with minimal downtime.

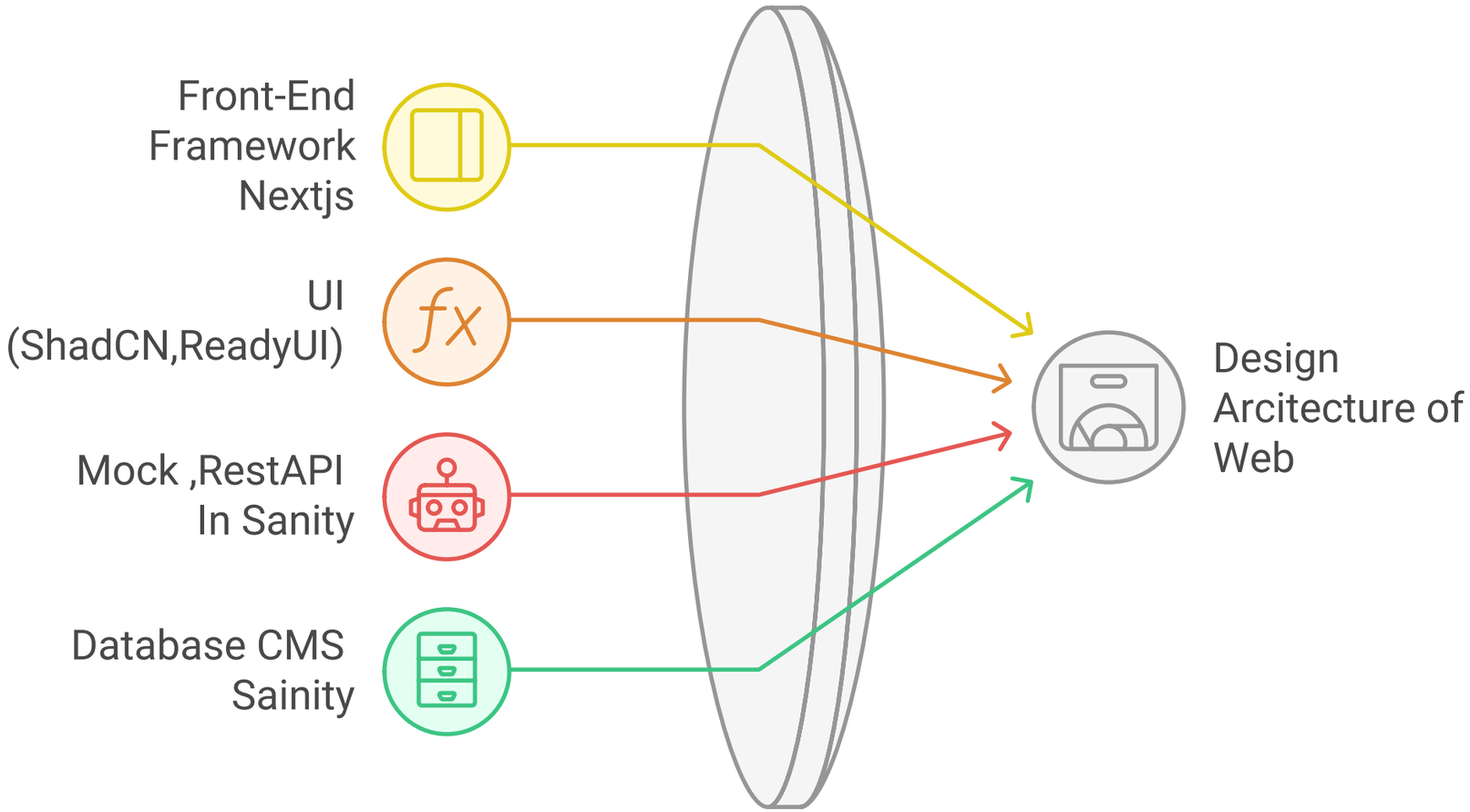


2. Design System Architecture

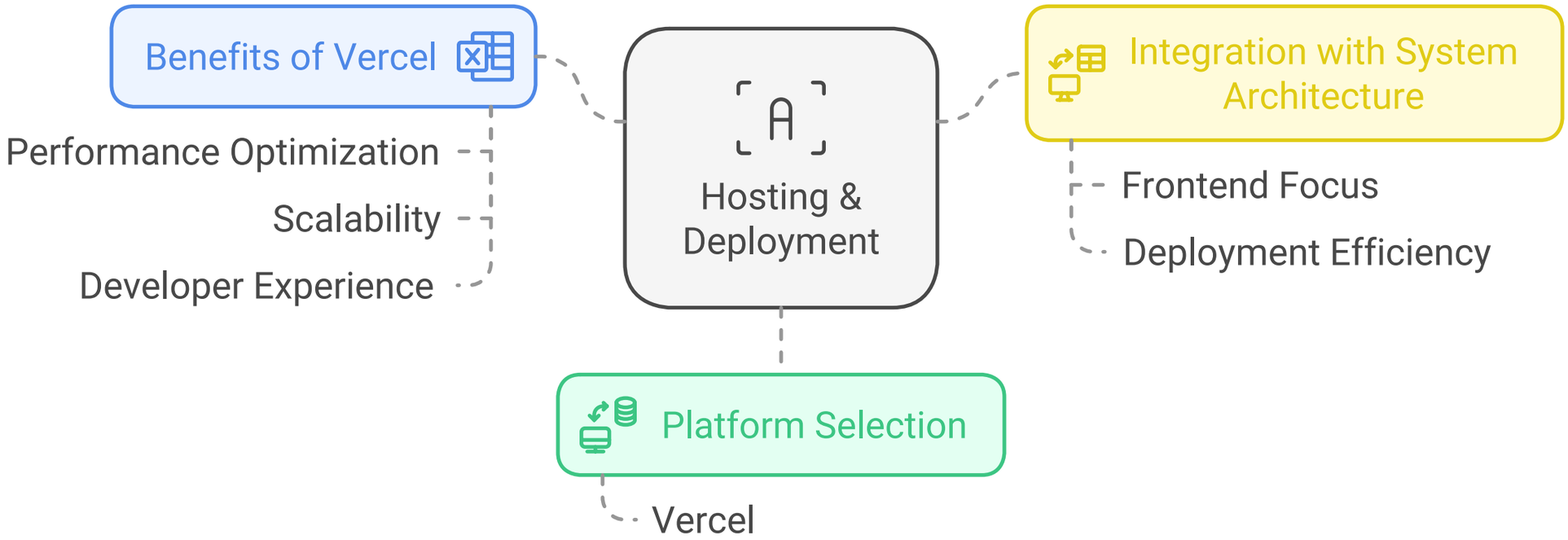
- **Front-End:**

    - Framework: Next.js.
    - UI/UX Library: Tailwind CSS, ShadCN, ReadyMadeUI, or Material-UI.
    - Features: Responsive design, intuitive navigation, and product reviews.
- **Back-End:**
    - Framework: Node.js with Mock API and Context API.
    - Features: Business logic, API handling, and database communication.
- **Database:**
    - Use Sanity as a headless CMS to manage content and product data.
    - Schema Fields:
        - **Products:** ID, name, price, stock, description, category, and images.
        - **Users:** ID, name, email, password, and contact info.
        - **Orders:** Order ID, user ID, product IDs, status, and total amount.
        - **Shipments:** Shipment ID, order ID, status, and delivery date.

Building a Cohesive Digital Platform



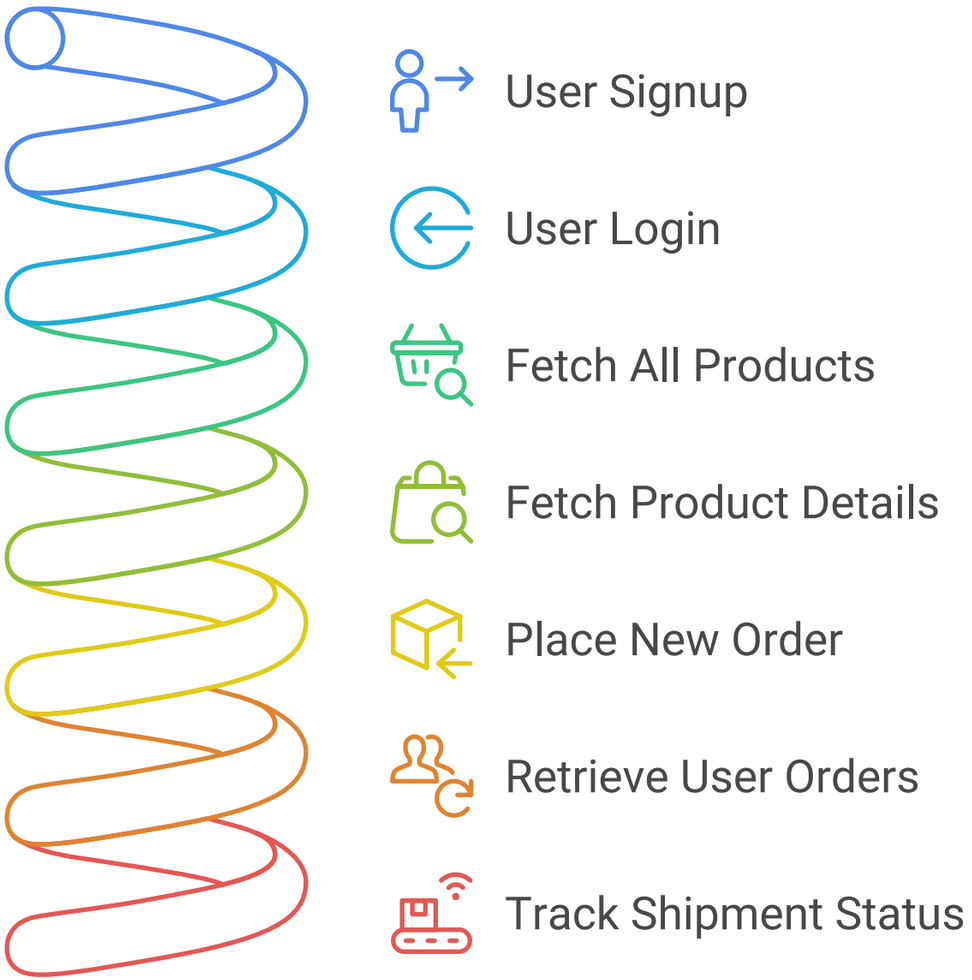
- **Hosting & Deployment:**
  - Platform: Vercel.



3. Plan API Requirements

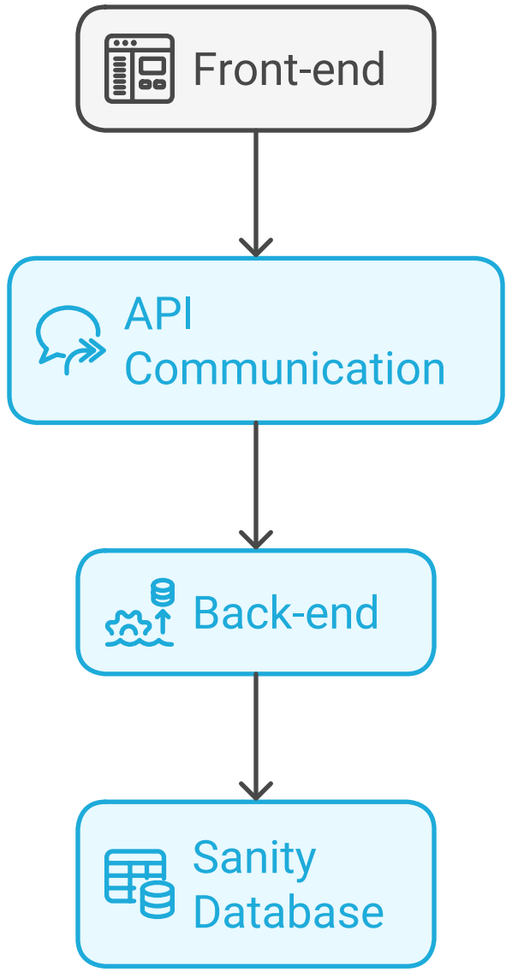
- **API Endpoints:**
    1. **User APIs:**
        - POST **/api/users/signup**: Create a new user account.
        - POST **/api/users/login**: User authentication.
    2. **Product APIs:**
        - GET **/api/products**: Fetch all products.
        - GET **/api/products/:id**: Fetch product details.
    3. **Order APIs:**
        - POST **/api/orders**: Place a new order.
        - GET **/api/orders/:userId**: Retrieve user orders.
    4. **Shipment APIs:**
        - GET **/api/shipments/:orderId**: Track shipment status.
- **Authentication:**
    - Optional JWT for token-based secure communication.
- **API Testing:**
    - Use Mock API to simulate and test the above endpoints during development.

API Interaction Sequence for Premii Woods



4. Write Technical Documentation

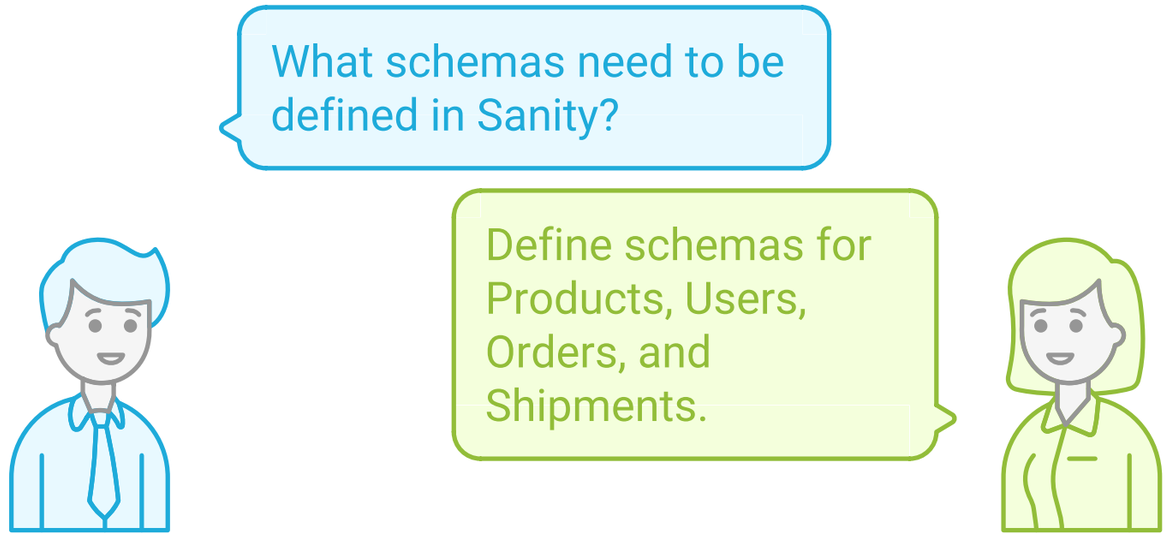
- **Architecture Diagram:**
    - Include a diagram showcasing:
        - Front-end communicating with the back-end via APIs.
        - Back-end connecting with the Sanity database.


Front-end
API
Communication
Back-end
Sanity
Database

- **Database Schema:**
    - Define schemas in Sanity for:
        - Products, Users, Orders, and Shipments.


```
{  "product": {
    "id": "P001", "name": "Ergonomic Computer Table",
    "price": 199.99,
    "stock": 20,
    "description": "A premium ergonomic productivity.",
    "category": "Computer Tables",
    "images": [
      "https:/product1-front.jpg",
    ]},
  "shipments": [
    {
      "shipmentId": "S001",
      "orderId": "O001",
      "status": "In Transit",
      "deliveryDate": "2025-01-2     },] }
```


Database Schema Definition



- **API Documentation:**
    - Use tools like **Postman** or **Swagger** to
    - document and test APIs.

Tool 1

Tool 2

Deliverables by End of Day 2

- **System Architecture Diagram:** Visual representation of the platform.
- **Database Schema:** Defined schemas for Sanity CMS.
- **API Documentation:** Ready-to-implement Mock API endpoints.
- **Technical Documentation:** Comprehensive details about workflows and design decisions.

Made by Tayyab Ilyas (Student Leader)