Creating a comprehensive property management web application with multiple portals requires careful planning, design, and development. Below is a detailed specification and project summary for the proposed web application.

**Project Summary:**

The Property Management Web Application aims to provide a centralized platform for landlords, tenants, and property managers to manage properties, streamline communication, and handle financial transactions. The application will have three distinct portals: Landlord Portal, Tenant Portal, and Property Manager Portal. It will be built using a PHP backend framework, ReactJS for the frontend, and PostgreSQL as the database.

**Features and Functionality:**

**Landlord Portal:**

1. Profile Management:
   1. Landlords can create and update their profiles.
   2. Profile includes full name, address, phone number, email address, and profile picture.
2. Property Management:
   1. Add, edit, and remove property details.
   2. Manage property addresses, type, bedrooms, bathrooms, square footage, year built, amenities, and geolocation.
3. Property Suite Details:
   1. Record details of each residential suite including bedrooms, bathrooms, kitchen, living room, etc.
4. Common Property Management:
   1. Maintain information about shared spaces within the residential building.
5. Repair and Maintenance:
   1. Keep track of repair and maintenance records.
   2. Create and manage job orders, assign tasks, and track progress.
6. Property Inspection:
   1. Schedule, conduct, and record property inspections.
   2. Plan necessary actions based on inspection results.
7. Utility Account Management:
   1. Manage utility accounts for each property.
8. Bookkeeping Records:
   1. Record various financial transactions including purchase, sales, petty cash, payments, receipts, deposits, etc.
9. Settings and Reports:
   1. Configure application settings.
   2. Generate income statements (profit and loss statements).

**Tenant Portal:**

1. Profile Management:
   1. Tenants can create and update their profiles.
   2. Profile includes full name, address, phone number, email address, and profile picture.
2. Rental Applications:
   1. Tenants can submit rental applications.
   2. Track the status of submitted applications.
3. Rent Payment:
   1. Make rent payments through direct or indirect methods.
4. Maintenance Requests:
   1. Tenants can submit maintenance requests.
5. Utility Usage:
   1. View utility usage and payment history.

**Property Manager Portal:**

1. Profile Management:
   1. Property managers can create and update their profiles.
   2. Profile includes full name, address, phone number, email address, and profile picture.
2. Property Management:
   1. Manage property details, suites, and common areas.
3. Job Order Management:
   1. Create, assign, and track job orders for repairs and maintenance.
4. Property Inspection:
   1. Schedule, conduct, and record property inspections.
5. Utility Account Management:
   1. Manage utility accounts for properties.
6. Tenant Management:
   1. Track property occupancy and vacancies.
   2. Manage tenant information.

**Technology Stack:**

* Backend Framework: PHP (Laravel or Symfony, for example)
* Frontend Framework: ReactJS
* Database: PostgreSQL

**Design and Development Guidelines:**

1. **User Authentication and Authorization:**
   1. Implement secure user registration, login, and password reset functionalities.
   2. Use role-based access control for different portal functionalities.
2. Database Design:
   1. Design a normalized database schema to store user profiles, property details, financial transactions, and other relevant data.
   2. Use proper indexing and foreign key relationships for efficient data retrieval.
3. Backend Development:
   1. Develop APIs using the chosen PHP backend framework to handle data processing, authentication, and authorization.
   2. Implement business logic for property management, financial transactions, maintenance, etc.
4. Frontend Development:
   1. Design a responsive and user-friendly interface using ReactJS.
   2. Implement portal-specific dashboards for landlords, tenants, and property managers.
   3. Utilize state management libraries like Redux for handling application state.
5. Map Integration:
   1. Integrate map APIs to display property geolocations.
6. File Upload and Storage:
   1. Implement secure file upload functionality for profile pictures, property images, and documents.
7. Financial Transactions:
   1. Implement transaction recording, validation, and tracking for various financial operations.
8. Reporting:
   1. Develop reporting functionalities to generate income statements and financial analysis reports.
9. Security and Data Privacy:
   1. Implement strong security measures to protect user data and financial information.
   2. Use encryption for sensitive data transmission and storage.
10. Testing and Quality Assurance:
    1. Conduct thorough testing, including unit testing and integration testing, to ensure functionality and reliability.
    2. Implement error handling and validation to provide a seamless user experience.
11. Deployment and Hosting:
    1. Deploy the application on a secure web server.
    2. Choose a hosting solution that supports the chosen technology stack.
12. Documentation:
    1. Provide comprehensive documentation for installation, setup, usage, and troubleshooting.

Conclusion:

The Property Management Web Application will provide an efficient and user-friendly platform for landlords, tenants, and property managers to manage properties, financial transactions, maintenance, and more. By following the outlined design and development guidelines, the application can be developed and deployed successfully, offering a comprehensive solution to address the needs of all stakeholders involved in property management.