# Week 1-2: Foundation & Domain Structure

# **Project Structure Overview**



## **Detailed File Structure**

# 1. Domains Directory

Each domain represents a core business capability and follows consistent internal structure.

src/domains/
threats/
components/
ThreatIcon.tsx
index.ts
— molecules/
index.ts
<del></del>
\to ThreatMatrix.tsx
ThreatIntelligencePanel.tsx
L index.ts
Lindex.ts
context/
│
ThreatProvider.tsx
index.ts
hooks/
useThreatData.ts
useThreatFilter.ts
wseMitreMapping.ts
index.ts
services/
│
MitreAttackService.ts
Lindex.ts
models/
\ Threat.ts
│
Lindex.ts
utils/
threatCalculations.ts
— mitreMapping.ts
index.ts
index.ts
requirements/
components/
RequirementStatus.tsx
PriorityBadge.tsx
ComplianceIcon.tsx
Lindex.ts

RequirementsTable.tsx
RequirementsMatrix.tsx
index.ts
Lindex.ts
context/
RequirementsProvider.tsx
Lindex.ts
Lindex.ts
Lindex.ts
models/
Lindex.ts
index.ts
Lagrandian index.ts
capabilities/
components/
— atoms/
CapabilityIcon.tsx
— MaturityBadge.tsx
ProgressIndicator.tsx
index.ts
molecules/
CapabilityCard.tsx
CapabilityMetrics.tsx
CapabilityFilter.tsx
index.ts
organisms/
│

1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
London index.ts
context/
CapabilitiesContext.tsx
CapabilitiesProvider.tsx
index.ts
hooks/
useCapabilities.ts
useCapabilityAssessment.ts
useMaturityScoring.ts
index.ts
services/
CapabilityService.ts
AssessmentService.ts
L index.ts
— models/
——Capability.ts
Assessment.ts
Lindex.ts
index.ts
index.ts
risks/
components/
RiskIndicator.tsx
SeverityBadge.tsx
Impacticon.tsx
index.ts
— molecules/
— organisms/
index.ts
L index.ts
context/
RiskContext.tsx
RiskProvider.tsx

London index.ts
hooks/
useRiskData.ts
Lindex.ts
services/
London index.ts
— models/
Risk.ts
RiskAssessment.ts
London index.ts
— utils/
Lagrandian index.ts
index.ts
evidence/
components/
FreshnessIndicator.tsx
index.ts
molecules/
London index.ts
organisms/
EvidenceLibrary.tsx
EvidenceChain.tsx
EvidenceAudit.tsx
Landindex.ts
index.ts
context/
EvidenceContext.tsx
useEvidence.ts
useEvidenceChain.ts
Graduate Chain.ts
services/
JULI SULUCIO

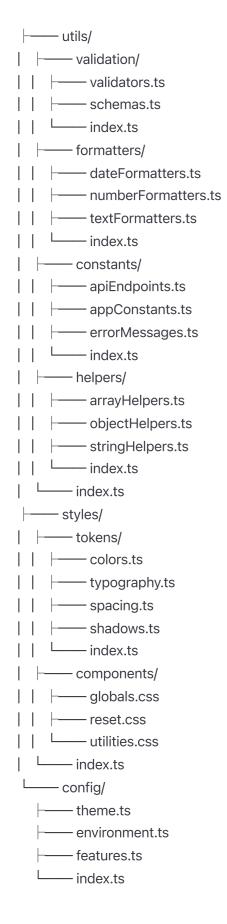
Lindex.ts
— models/
London index.ts
│ ├── utils/
Lindex.ts
Lindex.ts
├ trust/
components/
│
Lindex.ts
│
Lindex.ts
Lindex.ts
services/
│
│
Lindex.ts
— models/
— TrustMetrics.ts
L index.ts

rustCalculations.ts
confidenceModeling.ts
Lindex.ts
Lindex.ts
L resources/
components/
Lindex.ts
molecules/
Lagrandian index.ts
organisms/
ResourceSwimlane.tsx
London index.ts
Lindex.ts
context/
ResourceContext.tsx
ResourceProvider.tsx
index.ts
hooks/
useResources.ts
useResourcePlanning.ts
useCapacityPlanning.ts
index.ts
services/
ResourceService.ts
PlanningService.ts
index.ts
models/
Resource.ts
ResourcePlan.ts
index.ts
tils/
resourceCalculations.ts
capacityModeling.ts
index.ts
IIIUCX.LS

2. Shared Directory
Cross-cutting concerns used across domains.

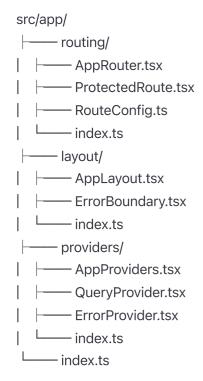
Button.tsx  Button.stories.tsx  Button.test.tsx  Index.ts
Button.stories.tsx  Button.test.tsx  Index.ts
Button.test.tsx
index.ts
Input.stories.tsx
Input.test.tsx
L index.ts
lcon/
— Icon.tsx
L index.ts
Badge.tsx
Badge.stories.tsx
index.ts
LoadingSpinner/
Tooltip/
— Avatar/
index.ts
molecules/
SearchBar/
SearchBar.tsx
index.ts
FilterDropdown/
—— Pagination/
DatePicker/
Modal/
Notification/
index.ts
— DataTable/
DataTable.tsx
DataTableColumn.tsx
DataTableRow.tsx
L index.ts
Navigation/
Sidebar.tsx
TopBar.tsx
Breadcrumbs.tsx

│
Lindex.ts
London index.ts
L index.ts
contexts/
AppContext.tsx
ThemeContext.tsx
AuthContext.tsx
NotificationContext.tsx
Lagrandian index.ts
hooks/
useLocalStorage.ts
useDebounce.ts
useToggle.ts
useAsync.ts
usePagination.ts
useSort.ts
Lindex.ts
services/
BaseApiService.ts
ApiClient.ts
— ApiTypes.ts
index.ts
storage/
LocalStorageService.ts
—— AuthService.ts
TokenService.ts
index.ts
notification/
Notification Service.ts
index.ts
index.ts
•



## 3. App Directory

Application shell and configuration.



## 4. Types Directory

Global TypeScript definitions.

```
src/types/

api.ts # API response/request types

common.ts # Shared utility types

entities.ts # Core business entity types

events.ts # Domain events

permissions.ts # Authorization types

index.ts
```

#### 5. Root Level Files

## **Implementation Order**

## **Day 1-2: Core Structure Setup**

- 1. Create the folder structure
- 2. Set up basic TypeScript configurations
- 3. Create initial index.ts files with proper exports

## **Day 3-4: Shared Foundation**

- 1. Implement base atomic components (Button, Input, Icon)
- 2. Set up theme and styling tokens
- 3. Create core contexts (App, Theme, Auth)

### **Day 5-6: Domain Scaffolding**

- 1. Create domain-specific folder structures
- 2. Define initial TypeScript interfaces for each domain
- 3. Set up basic service interfaces

### **Day 7-8: Context Architecture**

- 1. Implement domain-specific contexts
- 2. Set up data flow patterns
- 3. Create basic hooks for each domain

## **Day 9-10: Migration Planning**

- 1. Map existing components to new structure
- 2. Create migration scripts for moving code
- 3. Set up build and test configurations

#### This structure provides:

- Clear separation of concerns between domains
- Atomic design consistency across all components
- Scalable architecture for adding new features
- Type safety throughout the application
- Testability with isolated, mockable services
- Maintainability through consistent patterns

Would you like me to detail the specific implementation of any particular domain or shared component?