 created a comprehensive organization service structure with the following components:

1. **Models**:
   * **Organization**: Core organization entity with fields for tenant association, organization details, and settings
   * **Branch**: Branch/campus model with location, facilities, and capacity tracking
   * **Department**: Department model with hierarchical structure and resource management
2. **Controllers**:
   * **OrganizationController**: Handles organization CRUD and management operations
   * **BranchController**: Manages branch operations and analytics
   * **DepartmentController**: Handles department hierarchy and resource management
3. **Routes**:
   * RESTful API endpoints for organizations, branches, and departments
   * Proper authentication and authorization middleware
   * Validation middleware for request parameters
4. **Middleware**:
   * Organization-specific validation middleware
   * Access control for organizations, branches, and departments
   * Helper functions for checking user permissions
5. **Database Migrations**:
   * Organization table with tenant association and verification status
   * Branch table with facilities and capacity tracking
   * Department table with hierarchical structure support

The service is designed to be:

* **Scalable**: Proper separation of concerns and modular design
* **Secure**: Authentication, authorization, and input validation at multiple levels
* **Maintainable**: Clear code organization and comprehensive error handling
* **Extensible**: Easy to add new features and integrate with other services