@startuml Todo Management System Architecture

!define RECTANGLE class

title Todo Management System - Architecture Overview

package "Frontend (Next.js)" {
  RECTANGLE "Pages" as Pages {
    - Login/Register
    - Dashboard
    - Todos Management
    - File Management
    - Admin Panel
    - User Profile
  }

  RECTANGLE "Components" as Components {
    - Navigation
    - Todo Cards
    - File Upload
    - User Forms
    - Admin Tables
  }

  RECTANGLE "State Management" as State {
    - Auth Context
    - Todo Hooks
    - API Client
  }

  RECTANGLE "UI Library" as UI {

```
     - shadcn/ui

     - Tailwind CSS

     - React Hook Form

  }

}


package "Backend (Express.js)" {

  RECTANGLE "API Routes" as Routes {

    - /auth/*

    - /users/*

    - /todos/*

    - /files/*

  }


  RECTANGLE "Controllers" as Controllers {

    - AuthController

    - UserController

    - TodoController

    - FileController

  }


  RECTANGLE "Middleware" as Middleware {

    - Authentication

    - Authorization

    - Validation

    - Error Handling

    - Logging

  }


  RECTANGLE "Services" as Services {

    - JWT Service
```

```
      - File Service

      - Email Service

    }

  }


  package "Database Layer" {

    RECTANGLE "Prisma ORM" as Prisma {

      - Schema Definition

      - Query Builder

      - Migrations

      - Type Safety

    }


    database "PostgreSQL" as DB {

      - Users Table

      - Todos Table

      - Files Table

    }

  }


  package "File Storage" {

    folder "Local Storage" as Storage {

      - Profile Pictures

      - Todo Attachments

      - Temporary Files

    }

  }


  package "External Services" {

    cloud "Email Service" as Email {

      - Notifications
```

```
    - Password Reset

  }


  cloud "Cloud Storage" as Cloud {

    - File Backup

    - CDN Distribution

  }
}


' Relationships

Pages --> Components : uses

Components --> State : manages

State --> UI : styled with


Routes --> Controllers : handles

Controllers --> Middleware : protected by

Controllers --> Services : uses

Controllers --> Prisma : queries


Prisma --> DB : connects to

Controllers --> Storage : reads/writes

Services --> Email : sends

Storage --> Cloud : syncs to


' HTTP Communication

State ..> Routes : HTTP/REST API

Routes ..> State : JSON Response


' Data Flow

DB --> Prisma : SQL Queries

Prisma --> Controllers : Type-safe Data
```

Controllers --> Routes : JSON Response

Routes --> State : HTTP Response

State --> Components : React Props

Components --> Pages : Composition

note right of DB

  - User authentication data

  - Todo items and metadata

  - File references and metadata

  - Audit logs and timestamps

end note

note left of Storage

  - Secure file uploads

  - Access control

  - File type validation

  - Size limitations

end note

note top of Email

  - User notifications

  - Todo assignments

  - System alerts

  - Password resets

end note

@enduml