```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <title>Design - Study in Woods</title>
   <style>
       * { margin: 0; padding: 0; box-sizing: border-box; }
       @page { size: A4; margin: 1in; }
       body { font-family: 'Times New Roman', Times, serif; font-
size: 12pt; line-height: 1.6; color: #000; }
       h1 { text-align: center; font-size: 16pt; font-weight:
bold; text-decoration: underline; margin: 20px 0 30px 0; text-
transform: uppercase; }
       h2 { font-size: 14pt; font-weight: bold; margin-top: 25px;
margin-bottom: 15px; text-decoration: underline; }
       h3 { font-size: 12pt; font-weight: bold; margin-top: 20px;
margin-bottom: 10px; font-style: italic; }
       p { text-align: justify; margin-bottom: 12px; text-indent:
0.5in; }
       p.no-indent { text-indent: 0; }
       ul, ol { margin-left: 0.75in; margin-bottom: 12px; }
       li { margin-bottom: 8px; text-align: justify; }
       .page-break { page-break-after: always; }
       table { width: 100%; border-collapse: collapse; margin:
20px 0; }
       th, td { border: 1px solid #000; padding: 8px; text-align:
left; vertical-align: top; }
       th { background-color: #f0f0f0; font-weight: bold; }
       .diagram-box { border: 2px solid #000; padding: 15px;
margin: 20px 0; font-family: 'Courier New', monospace; font-size:
9pt; background-color: #f9f9f9; white-space: pre-wrap; }
   </style>
</head>
<body>
   <h1>6. DESIGN</h1>

   <h2>6.1 System Architecture</h2>

   <p>The Study in Woods platform implements a three-tier
architecture consisting of Presentation Layer (Next.js frontend),
Application Layer (Go Fiber backend API), and Data Layer
(PostgreSQL database, Redis cache, DigitalOcean Spaces). This
separation enables independent scaling, technology replacement,
and clear responsibility boundaries. Communication between tiers
occurs through well-defined RESTful APIs with JSON payloads,
Server-Sent Events for real-time streaming, and S3-compatible
protocols for file storage.</p>

   <div class="diagram-box">



   Pages        Components       Hooks          State
 (Routing)        (shadcn)     (React Hook)   (TanStack)




  Handlers        Services      Middleware      Utils
 (API Routes)  (Business        (Auth, CORS)  (Crypto,
                Logic)                         Logger)


PostgreSQL 15       Redis 7      DO Spaces      DO GradientAI
(Primary Data)       (Cache)      (S3 Storage)  (LLM/RAG)
Sessions     Llama 3.3    Rate Limit  Knowledge      Pub/Sub

   </div>

   <h2>6.2 Data Flow Diagrams</h2>

   <h3>6.2.1 Level 0 - Context Diagram</h3>

   <div class="diagram-box">

  Login, Upload PDFs, Ask Questions,
  Study in      User        Woods
  Authentication Tokens, PDF Parsing,
              AI Responses, Analytics

  Manage Users, Configure System,                      Admin
User
```

```
   </div>

   <h3>6.2.2 Level 1 - System Decomposition</h3>

   <div class="diagram-box">
Student > [1.0 Authentication] > User Database
             > [2.0 Academic Management] > 
University/Course/Subject DB

             > Upload PDFs
             > Browse Hierarchy
             > [3.0 Document Processing] > Documents DB + DO 
Spaces

             > Extract Syllabus > AI Service
             > Index to Knowledge Base
             > [4.0 AI Chat System] > ChatSessions + ChatMessages 
DB

             > Send Message
             > [5.0 Analytics] > Activity Logs + Usage Stats DB

Admin > [6.0 Administration] > All Databases + Audit Logs
          > User Management
          > View Audit Logs
   </div>

   <h3>6.2.3 Level 2 - Process Detail: Document Upload & 
Processing</h3>

   <div class="diagram-box">
[3.0 Document Processing]
   > [3.1 Validate Upload]
    Input: File (from User)
    Process: Check file type (PDF), size (<10MB), duplicate
    Output: Validation Result

                       > [3.3 Save Metadata]
    Input: File metadata, Storage info
    Process: Insert Document record with status='pending'
    Output: Document ID
    Database: Documents table

                            > [3.5 Store Syllabus]
    Input: Extracted JSON
    Process: Parse and insert into Syllabus, SyllabusUnit, 
```

SyllabusTopic

Database: Syllabus tables (3 tables)

Input: Document ID, PDF URL

Update: Document.indexing_status = 'completed'

~~Course Subject, with documents and chat sessions associated with subjects. User authentication and activity tracking tables support system security and analytics. Additional tables include chat memory management (chat_memory_batches, chat_compacted_contexts), PYQ paper management (pyq_papers, pyq_questions), syllabus structure (syllabi, syllabus_units, syllabus_topics), background processing (indexing_jobs), and user engagement (user_notifications).</p>~~

~~<div class="diagram-box">~~

~~1:N~~

~~FK university_id~~

~~name~~

~~duration~~

~~1:N~~

~~FK course_id~~

~~Subject~~

~~PK id~~

~~code~~

~~1:N~~ ~~ChatSession~~

~~Document PK id PK id FK user_id FK subject_id FK subject_id filename title spaces_url status 1:N~~

~~1:N~~

~~ChatMessage Syllabus PK id FK~~

~~session_id PK id FK user_id FK subject_id role~~
~~FK document_id~~

~~Other Tables:~~
~~SyllabusUnit APIKeyUsageLog~~
~~PK id AdminAuditLog AppSetting~~
~~title PYQ (Questions)~~

~~</div>~~

~~<h2>6.4 Sequence Diagrams</h2>~~

~~<h3>6.4.1 User Login Flow</h3>~~

~~<div class="diagram-box">~~
~~User Frontend API Handler Auth Service~~
~~Database Redis~~

~~Login>~~ ~~(email,pwd)~~
~~POST /login>~~
~~Validate~~

~~User Record~~
~~<~~

~~Check Rate>~~
~~OK/Block~~
~~Generate JWT~~
~~Store Session~~

~~JWT Token~~
~~<~~
~~<~~

~~</div>~~

~~<h3>6.4.2 Document Upload & Syllabus Extraction Flow</h3>~~

~~<div class="diagram-box">~~

~~User Frontend API Syllabus DO Spaces Database AI~~
~~Service KB API~~
~~> PDF~~
~~POST~~
~~Validate~~
~~Upload~~
~~<~~
~~Save~~
~~<~~
~~Accepted~~
~~[Background Job]~~
~~Extract>~~
~~(Llama 3.3)~~
~~syllabus as JSON <JSON~~
~~(Syllabus tables)~~
~~<Completed~~
~~>~~
~~<~~
~~Progress~~
~~Ask>~~
~~POST~~
~~/chat/msg~~
~~Last 10 Messages~~
~~Assemble Context~~
~~(System prompt +~~
~~conversation history)~~
~~(user msg)~~
~~Top 5 docs~~
~~Chat Request~~
~~(streaming)~~
~~<Tokens~~
~~Stream (word by word)~~
~~>~~

~~Messages (user + assistant,~~
~~citations)~~
~~<~~
~~</div>~~

~~<div class="page-break"></div>~~

~~<h2>6.5 Component Diagrams</h2>~~

~~<h3>6.5.1 Backend Component Architecture</h3>~~

~~<div class="diagram-box">~~
~~API Layer~~

~~JWT CORS Rate Limit Logger~~
~~Verify Policy (Redis)~~

~~Service Layer~~
~~Repository Layer~~
~~Data Layer~~

~~Stats Cards~~
~~Quick Actions~~

~~/universities~~
~~University Form (Create/Edit)~~

~~/courses~~
~~Course Form~~

~~/subjects~~
~~Subject Detail~~
~~Session List (Sidebar)~~
~~User Message~~
~~Citation Panel~~

~~/analytics~~
~~Statistics Cards~~

~~API Client (Axios)~~
~~Validators (Zod schemas)~~

~~   </div>~~

~~   <div class="page-number">6</div>~~
~~</body>~~
~~</html>~~