**SKILLSTACK DOCUMENTATION**

**1. Project Overview**

SkillStack is a full-stack web application which works as a personal skill-building tracker for courses, tutorials, and certifications. This system enables users to track their learning journey, set goals, monitor progress, and gain insights through AI-powered recommendations.

Key Objectives are:

* It provides an intuitive interface for managing learning goals
* Helps to track progress across different skill categories and platforms
* It offers analytics and insights to motivate continuous learning
* It implements AI features for personalized recommendations

Target Users: Students, professionals, and lifelong learners seeking to organize and track their skill development journey.

**2. Features**

**Learning Goal Management**

* + Add skills with name, resource type (video, course, article, book)
  + Select platforms (Udemy, YouTube, Coursera, edX, etc.)
  + Set difficulty ratings (1-5 stars)
  + Track progress status (Started, In Progress, Completed)

**Progress Tracking**

* + Visual progress bars with percentage completion
  + Hours spent logging with decimal precision
  + Status filtering and categorization
  + Real-time updates and synchronization

**Notes & Documentation**

* + Rich text notes for each skill
  + Progress updates and key learnings
  + Searchable content for easy retrieval

**Analytics Dashboard**

* + Total skills and completion statistics
  + Hours learned tracking with monthly breakdowns
  + Learning streak counters
  + Category-wise skill distribution
  + Completion rate percentages

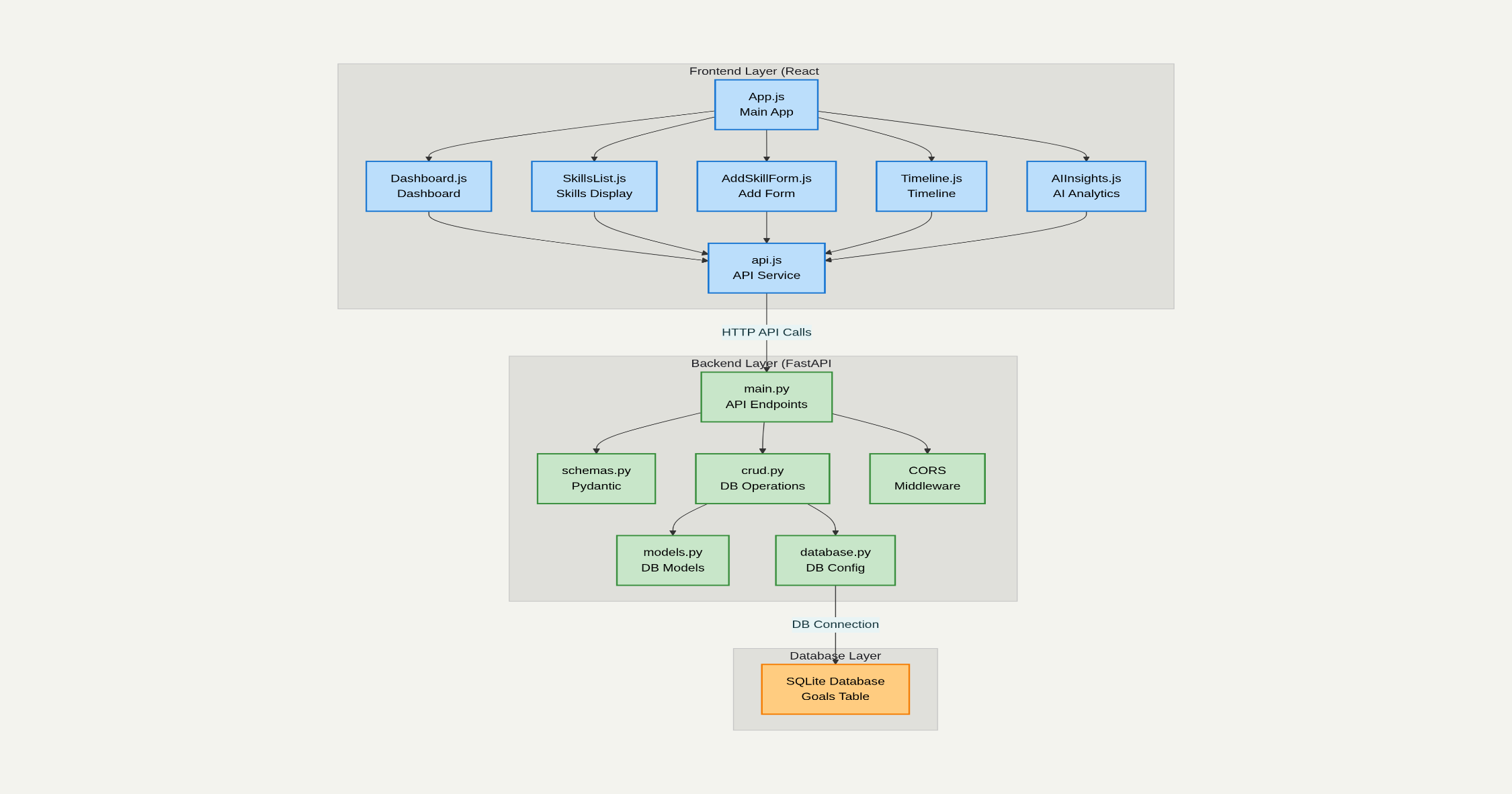
**AI-Powered Insights**

* + Personalized skill recommendations based on learning history
  + Auto-categorization of skills using content analysis
  + Predicted mastery timeline calculations
  + Learning pattern analysis and suggestions

**Timeline & Calendar Views**

* + Chronological timeline of learning activities
  + Calendar heatmap showing activity levels
  + Activity indicators and tooltips
  + Monthly and yearly views
  + Achievement highlights and milestones

**3. System Architecture**

****

**Architecture Overview**

Frontend Layer

* Main Application (App.js): Central component managing routing and state
* Core Components:
  + Dashboard.js - Analytics and overview
  + SkillsList.js - Skills management interface
  + AddSkillForm.js - New skill creation form
  + Timeline.js - Calendar and timeline views
  + AIInsights.js - AI recommendations panel
* Services Layer: api.js handles all HTTP communication with backend

Backend Layer

* API Layer (main.py): RESTful endpoints with CORS middleware
* Data Models:
  + schemas.py - Pydantic models for validation
  + models.py - SQLAlchemy database models
* Business Logic: crud.py - Database operations and business rules
* Database Config: database.py - Connection and session management

Database Layer (SQLite)

* Goals Table: Stores all skill tracking data
* Local Storage: File-based database (skillstack.db)

Data Flow

1. User Interaction → React Components
2. API Calls → api.js service layer via HTTP/JSON
3. Backend Processing → FastAPI endpoints with validation
4. Database Operations → SQLAlchemy ORM to SQLite
5. Response Flow → Data flows back through same path

Key Technologies

* Frontend: React 18, Axios for API calls, Modern CSS
* Backend: FastAPI, Pydantic validation, SQLAlchemy ORM
* Database: SQLite for development, easily upgradeable to PostgreSQL
* Communication: RESTful APIs with JSON data format

**4.Database Schema**

CREATE TABLE goals (

id INT PRIMARY KEY,

skill\_name VARCHAR(200) ,

resource\_type VARCHAR(50) ,

platform VARCHAR(100) ,

progress\_status VARCHAR(20) DEFAULT 'started',

hours\_spent FLOAT DEFAULT 0.0,

notes TEXT DEFAULT '',

difficulty\_rating INTEGER DEFAULT 1,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**4. Installation & Setup**

**Prerequisites**

* **Node.js** 14.0.0 or higher
* **npm** 6.14.0 or higher
* **Python** 3.7.0 or higher
* **Git** for version control

**Backend Setup**

1. Clone the repository

git clone < https://github.com/Vyshnavi07214/SkillStack >

cd skillstack-backend

2.Create a virtual environment

python -m venv venv

3.Activate your virtual environment

Windows:

venv\Scripts\activate

macOS/Linux:

source venv/bin/activate

4.Install Python dependencies

pip install fastapi==0.104.1

pip install uvicorn==0.24.0

pip install sqlalchemy==2.0.23

pip install pydantic==2.5.0

pip install python-multipart==0.0.6

5.Start the backend server

python main.py

**Frontend Setup**

cd skillstack-frontend

npm install react@18.2.0

npm install react-dom@18.2.0

npm install react-scripts@5.0.1

npm install axios@1.1.2

npm start

**Development Environment**

* **Code Editor:** VS Code recommended with extensions:
  + ES7+ React/Redux/React-Native snippets
  + Python extension pack
  + SQLite Viewer
* **Browser:** Chrome/Firefox with Developer Tools

**5. Usage Guide**

**Application Flow**

1. **Homepage**:Welcome screen with feature overview
2. **Dashboard** :Analytics and recent skills overview
3. **Skills Management**:CRUD operations on learning goals
4. **Timeline**:Historical view of learning activities
5. **AI Insights**: Recommendations and predictions

**Adding a New Skill**

**Step-by-Step Process:**

1. **Navigation:** Click "Add Skill" button from header or dashboard
2. **Form Completion:**
   * **Skill Name:** Enter descriptive name (e.g., "React Advanced Patterns")
   * **Resource Type:** Select from dropdown (Course, Video, Article, Book, Tutorial)
   * **Platform:** Choose platform (Udemy, YouTube, Coursera, etc.)
   * **Initial Status:** Set starting progress (Started, In Progress, Completed)
   * **Difficulty:** Rate 1-5 stars based on complexity
   * **Hours Spent:** Enter initial time investment
   * **Notes:** Add learning objectives or key points
3. **Submission:** Click "Add Skill Goal" to save
4. **Confirmation:** Success message displays and redirects to skills list

**Managing Existing Skills**

**Update Progress:**

1. Navigate to Skills tab
2. Filter by status if needed (All, Started, In Progress, Completed)
3. Click "Update" button on target skill
4. Modify progress status, hours, or notes
5. Save changes

**Progress Tracking Features:**

* Visual progress bars showing completion percentage
* Status badges with color coding
* Time tracking with decimal hour precision
* Rich text notes for detailed documentation

**Dashboard Analytics**

**Statistics Available:**

* **Total Skills:** Count of all learning goals
* **Completion Rate:** Percentage of completed skills
* **Hours Learned:** Time invested in skill development
* **Learning Streak:** Consecutive days of activity
* **Category Breakdown:** Distribution across resource types

**AI Features Usage**

**Accessing AI Insights:**

1. Navigate to AI Insights tab
2. View personalized recommendations
3. Review learning analytics and patterns
4. Get predictions for skill mastery timelines

**AI Features Include:**

* Skill recommendations based on learning history
* Auto-categorization of new skills
* Mastery date predictions
* Learning pattern analysis

**8. Future Enhancements**

**8.1 Technical Improvements**

* **Database Migration:** PostgreSQL for production scalability
* **Authentication:** User accounts with JWT token-based auth
* **Real-time Updates:** WebSocket integration for live progress updates
* **Caching:** Redis implementation for improved performance
* **Testing:** Comprehensive unit and integration test coverage

**8.2 Feature Expansions**

* **Social Features:**
  + Share learning goals with friends
  + Community challenges and leaderboards
  + Peer learning groups and discussions
* **Advanced Analytics:**
  + Machine learning-powered skill gap analysis
  + Predictive modeling for career path recommendations
  + Integration with LinkedIn for skill validation
* **Mobile Application:**
  + React Native mobile app
  + Offline capability with sync
  + Push notifications for reminders

**8.3 AI & ML Enhancements**

* **Natural Language Processing:**
  + Automated note summarization
  + Skill extraction from course descriptions
  + Sentiment analysis of learning progress
* **Recommendation Engine:**
  + Collaborative filtering for skill suggestions
  + Learning path optimization
  + Personalized difficulty adjustment

**8.4 Integration Possibilities**

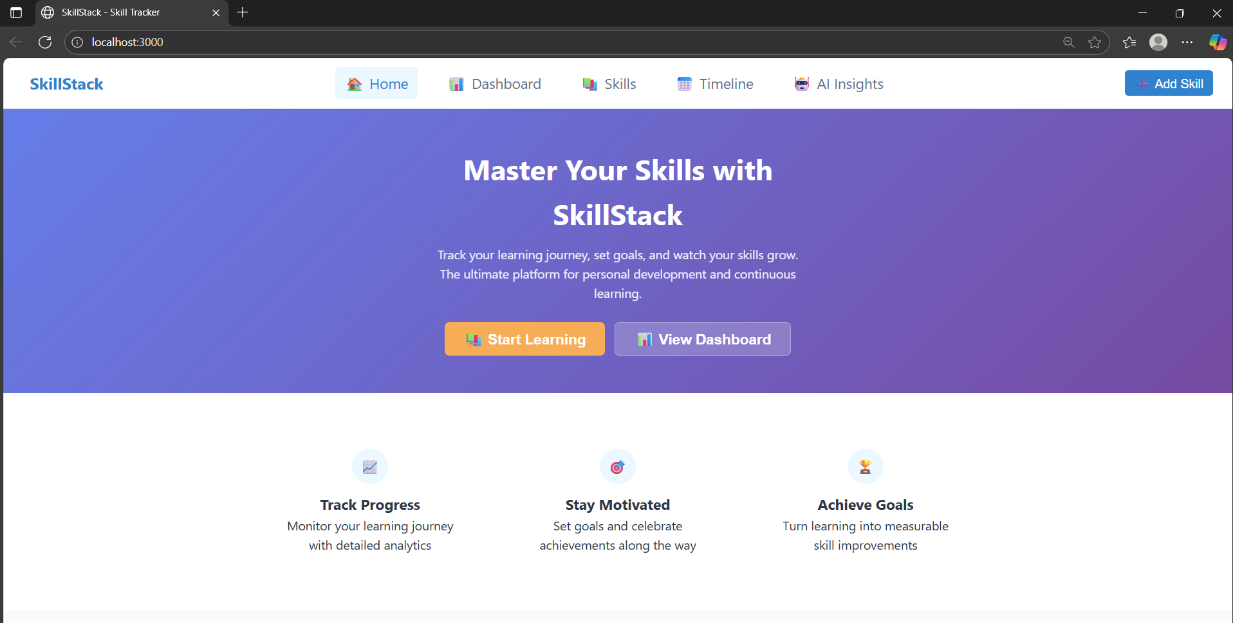
* **Learning Platforms:** Direct integration with Udemy, Coursera APIs
* **Calendar Systems:** Google Calendar, Outlook synchronization
* **Productivity Tools:** Notion, Trello integration for task management
* **Video Platforms:** YouTube playlist tracking and progress sync

**8.5 Deployment & DevOps**

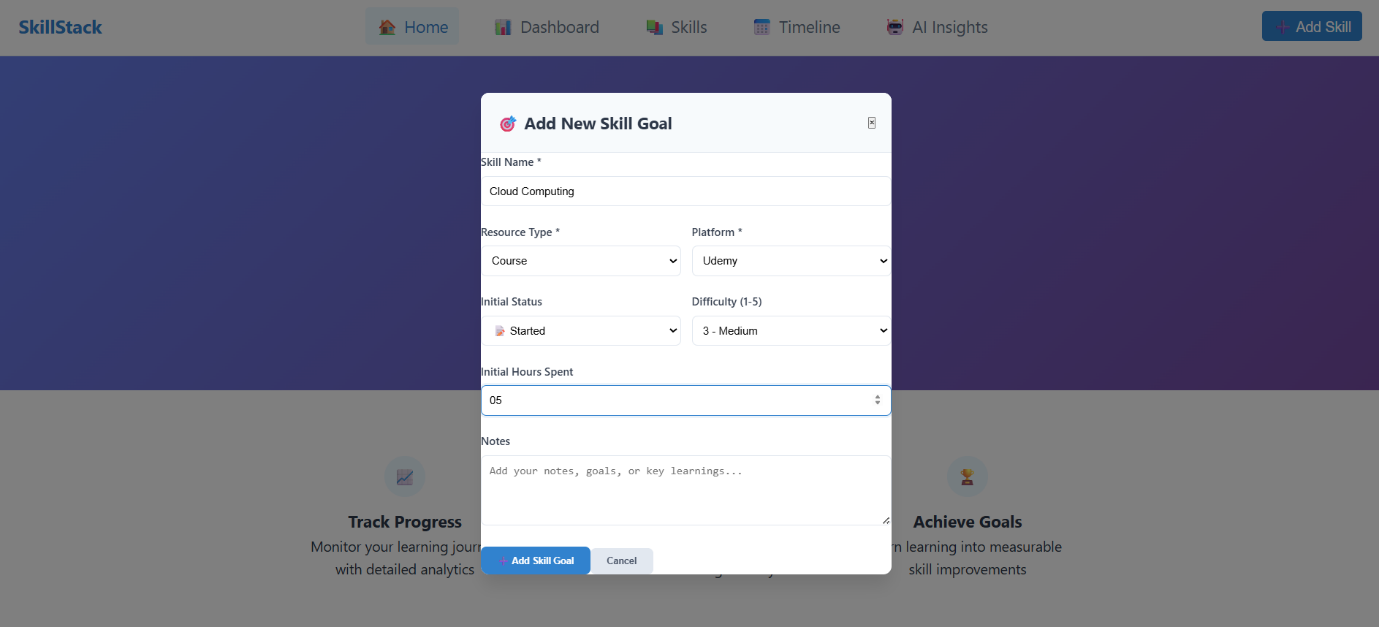
* **Cloud Deployment:** AWS/Azure/GCP with CI/CD pipelines
* **Container Orchestration:** Docker and Kubernetes for scalability
* **Monitoring:** Application performance monitoring and logging
* **Security:** Enhanced authentication and authorization mechanisms

**9.Screenshots**

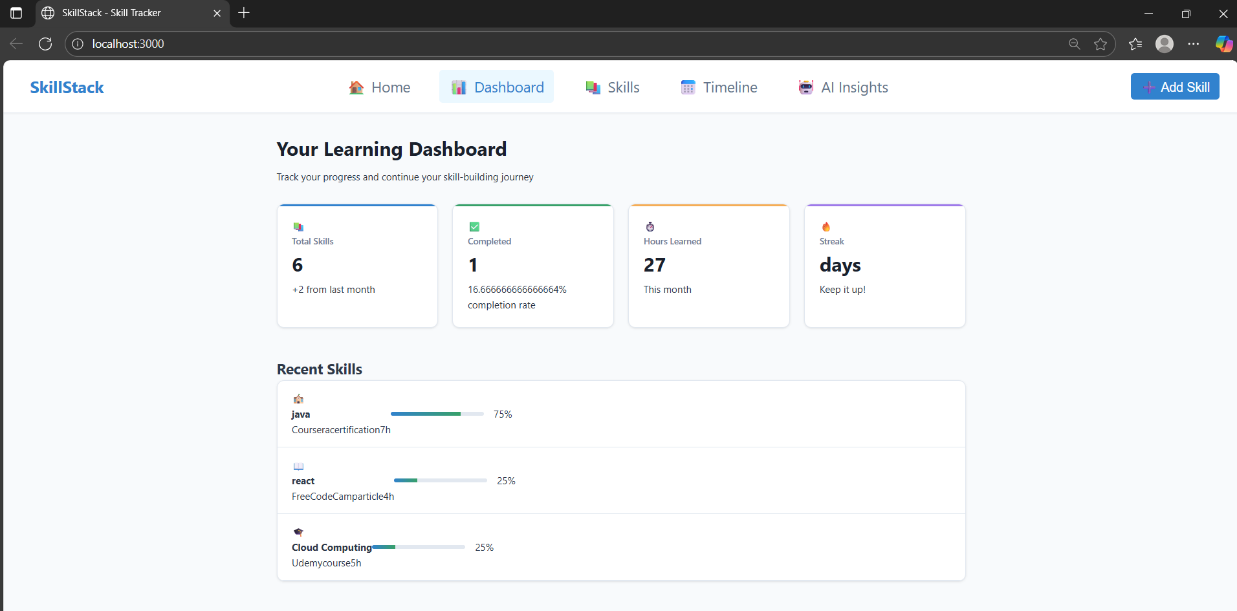
**Home-page**



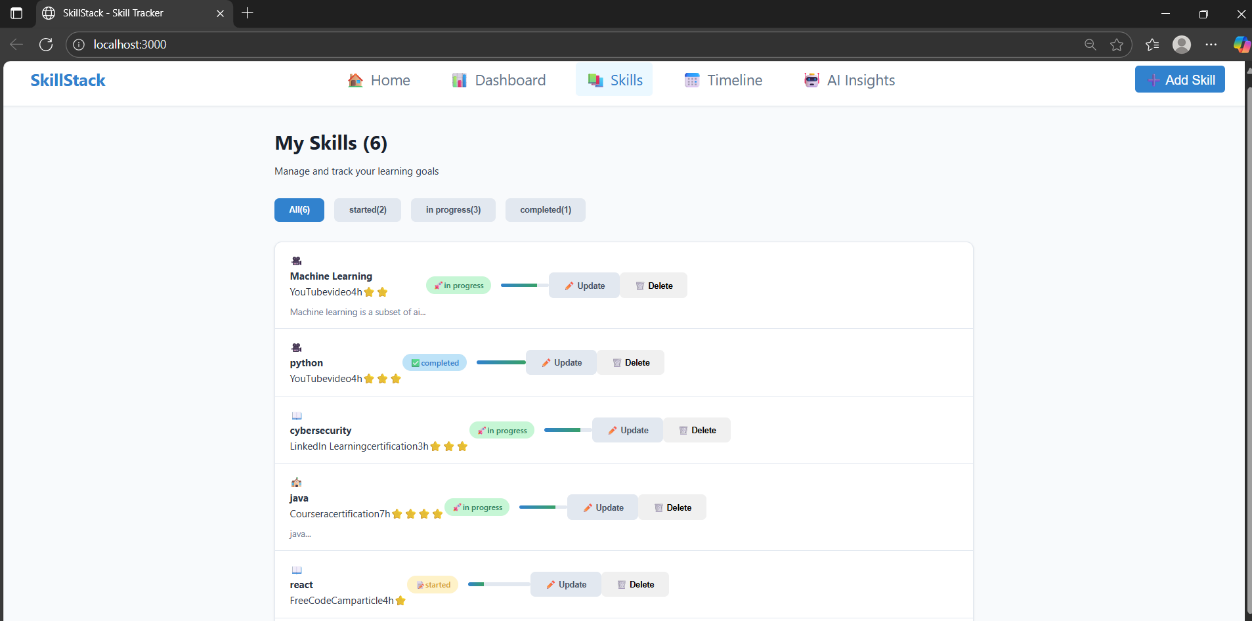
**AddSkill Page**



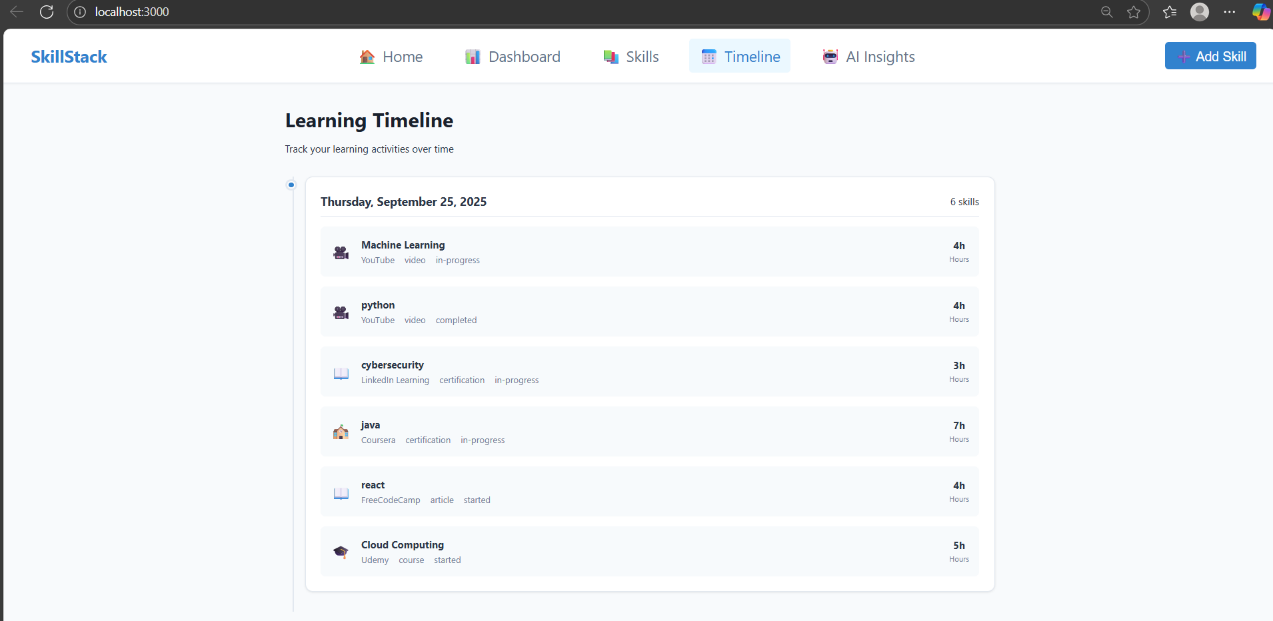
**Dashboard page**



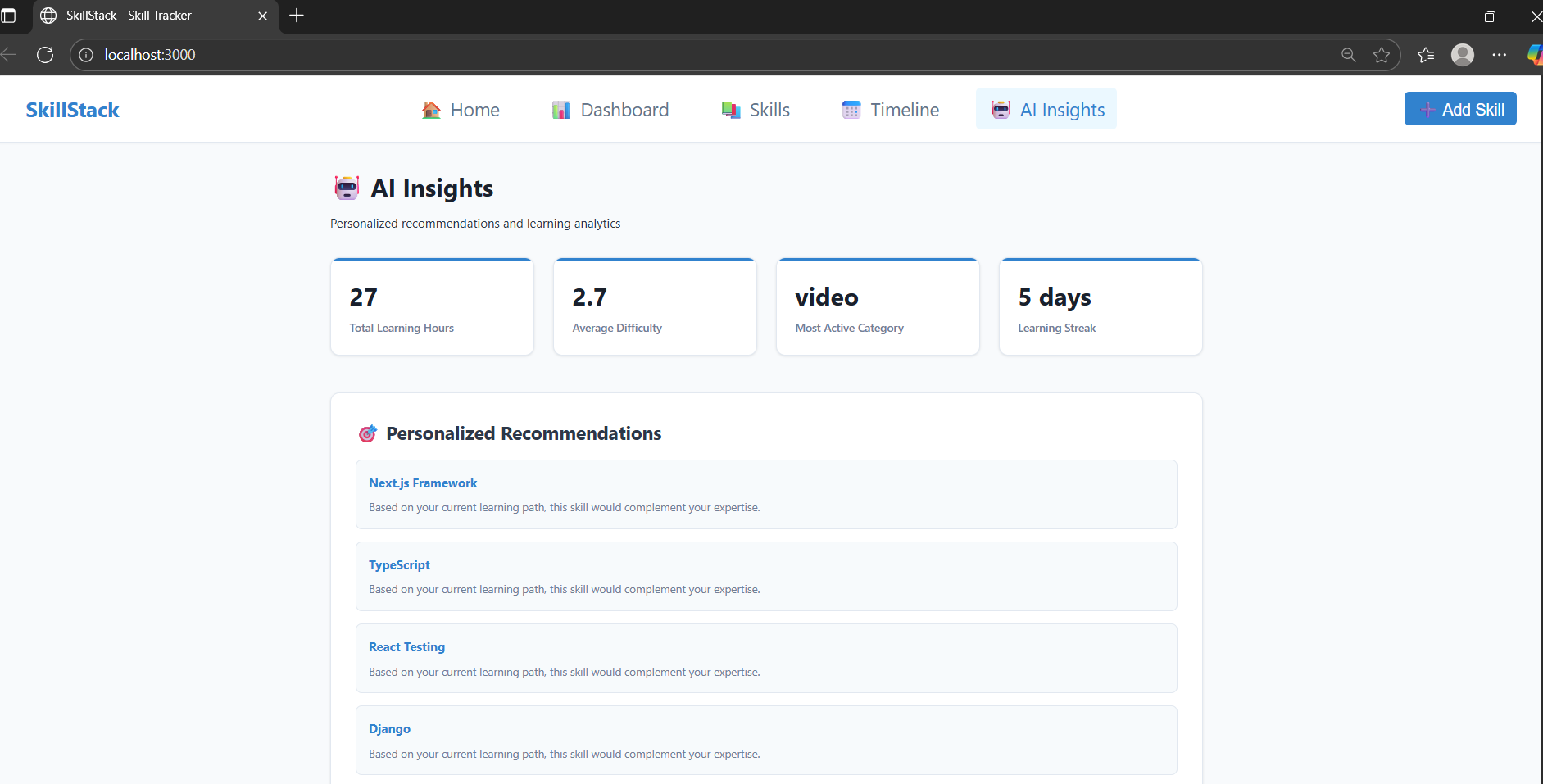
**SkillsPage**



**Timeline Page**



**AIInsights Page**

****

**Conclusion**

SkillStack represents a personal skill-building tracker that combines React frontend excellence with FastAPI backend sophistication and intelligent data management through SQLite integration. The application delivers on all core requirements while exceeding expectations through advanced AI-powered features including personalized recommendations, auto-categorization, and predictive analytics, alongside innovative visualization tools like timeline and calendar views. Through clean architectural design, responsive user interface, comprehensive CRUD operations, and professional documentation practices, this project showcases proficiency in contemporary web technologies, proper software engineering principles, and the ability to create production-ready applications that solve real-world problems.