AI Avatar Chat App: 5-Class Course Syllabus (2.5 hours per class)

I'll structure this syllabus into 5 sessions of 2.5 hours each, creating an engaging and comprehensive learning experience while maintaining a practical pace for students.

Class 1: AI Foundations & Python Essentials (2.5 hours)

Hour 1: Introduction to AI and Large Language Models (75 min)

- **Welcome and Course Overview** (15 min)
  - Introduction to the final project: AI Avatar Chat Application
  - Overview of the 5-class structure and learning objectives
  - Brief demonstration of the completed app

- **AI and LLM Fundamentals** (30 min)
  - Evolution of AI and the breakthrough of transformer architecture
  - How attention mechanisms revolutionized language models
  - Interactive demonstration: Claude capabilities and limitations
  - Visualization of how attention mechanisms focus on text

- **The Science of Digital Avatars** (30 min)
  - Overview of diffusion models and neural rendering techniques
  - Understanding 3D face modeling and audio-to-expression mapping
  - Visual demonstration of avatar generation technology
  - Ethics discussion: Responsible use of AI avatars

Hour 2: Python Programming for AI Applications (75 min)

- **Python Environment Setup** (20 min)
  - Installing necessary tools and libraries
  - Understanding virtual environments
  - Package management with pip and conda

- **Python Programming Essentials** (40 min)

  - Data structures and control flow
  - Functions and error handling
  - Working with files and JSON data
  - Hands-on coding: Building a simple text chatbot

- **Preview of Next Class and Setup Verification** (15 min)
  - Ensure all students have working environments
  - Overview of web application development with Streamlit
  - Assignment: Complete a simple Python exercise before next class

Class 2: Web Applications & API Integration (2.5 hours)

Hour 1: Building Interactive Web Apps with Streamlit (75 min)

- **Introduction to Web Application Concepts** (15 min)
  - Web application architecture basics
  - Frontend vs. backend development
  - Introduction to Streamlit for rapid application development

- **Streamlit Fundamentals** (30 min)
  - Setting up a Streamlit project
  - Creating interactive UI elements
  - State management and session handling
  - Data visualization capabilities

- **Building a Basic Chat Interface** (30 min)
  - Designing the user interface
  - Implementing chat history
  - Adding user input and response areas
  - Styling the application

Hour 2: Working with APIs (75 min)

- **API Fundamentals** (20 min)
  - Understanding RESTful APIs
  - Authentication and API keys
  - Making HTTP requests with Python

  - Handling API responses and errors

- **Working with the Claude API** (35 min)
  - Setting up the Claude client
  - Designing effective prompts
  - Processing and displaying AI responses
  - Implementing conversation history

- **Hands-on Exercise: API Integration** (20 min)
  - Students integrate Claude API into their Streamlit app
  - Testing different prompts and responses
  - Debugging common API issues

Class 3: Avatar Technology & Video Generation (2.5 hours)

Hour 1: Understanding Avatar Technology (75 min)

- **Deep Dive into Digital Human Technology** (30 min)
  - How diffusion models generate realistic images
  - Neural rendering techniques for photorealistic avatars
  - Voice synthesis and lip synchronization
  - Visual demonstration of the generation pipeline

- **Working with our Avatar API** (25 min)
  - API structure and documentation
  - Authentication and request formats
  - Understanding response handling
  - Managing asynchronous video generation

- **Visualizing the Video Generation Process** (20 min)
  - Step-by-step breakdown of avatar creation
  - Performance considerations and optimization
  - Quality vs. speed tradeoffs

Hour 2: Implementing Avatar Integration (75 min)

- **Building the Avatar Generator Component** (35 min)

  - Implementing the video generation function
  - Creating progress indicators
  - Handling API responses and errors
  - Displaying videos in the Streamlit interface

- **Avatar Selection and Customization** (20 min)
  - Creating the avatar selection sidebar
  - Implementing avatar style options
  - Preview functionality

- **Testing and Troubleshooting** (20 min)
  - Hands-on testing of avatar generation
  - Common errors and how to fix them
  - Performance optimization techniques

Class 4: Application Integration & Advanced Features (2.5 hours)

Hour 1: Bringing Everything Together (75 min)

- **Application Architecture Review** (15 min)
  - Review of components built so far
  - Data flow between components
  - Planning the integration approach

- **Full Application Integration** (40 min)
  - Connecting the chat interface with the Claude API
  - Linking Claude responses to avatar generation
  - Implementing the complete messaging flow
  - Building a cohesive user experience

- **Testing the Complete Application** (20 min)
  - End-to-end testing of the application
  - Identifying and fixing integration issues
  - Performance analysis

Hour 2: Adding Advanced Features (75 min)

- **Conversation History and Context** (25 min)
    - Implementing memory for the AI assistant
    - Managing conversation context
    - Enhancing responses with historical information

- **Error Handling and Fallbacks** (25 min)
    - Implementing robust error handling
    - Creating graceful fallback mechanisms
    - Rate limiting and quota management

- **UI Enhancements and Styling** (25 min)
    - Polishing the user interface
    - Adding animations and transitions
    - Responsive design for different devices

Looking at the current syllabus for Class 5, it doesn't explicitly structure the final class as a group project with avatar demos. Let me revise the final class to ensure it focuses on group-based avatar demos with three students per group.

Class 5: Group Avatar Projects & Demonstrations (2.5 hours)

Hour 1: Group Project Workshop (75 min)

- **Group Formation and Planning** (15 min)
    - Students divide into groups of three
    - Each group assigns roles (e.g., UI designer, API integrator, presentation lead)
    - Groups outline their unique avatar application concept
    - Planning worksheet for project features and division of work

- **Collaborative Development** (45 min)
    - Groups work together to enhance their avatar applications
    - Each student contributes to their assigned area
    - Implement at least one unique feature per group
    - Instructor circulates to provide guidance and troubleshooting

- **Preparation for Demonstrations** (15 min)
  - Groups prepare their presentation flow
  - Test their application for the demonstration
  - Prepare to explain their technical implementation
  - Create a brief "pitch" for their avatar concept

Hour 2: Avatar Project Demonstrations (75 min)

- **Group Presentations** (45 min)
  - Each group presents their avatar application (5-7 minutes per group)
  - Live demonstration of their working application
  - Explanation of technical challenges and solutions
  - Highlight unique features they implemented
  - Brief Q&A after each presentation

- **Peer Feedback and Voting** (15 min)
  - Students provide structured feedback to other groups
  - Vote for different categories: Most Creative, Best Technical Implementation, Best User Experience, Most Realistic Avatar

- **Course Wrap-up and Next Steps** (15 min)
  - Announcement of "winners" in each category
  - Review of key concepts learned throughout the course
  - Information about your premium avatar platform subscription
  - Resources for further learning and development
  - Final thoughts and course conclusion

Group Project Requirements

Each group of three students should:

1. **Create a Unified Application**
   - Combine their individual components into one cohesive application
   - Implement a consistent design theme
   - Ensure all parts work together smoothly

2. **Add Unique Features** (at least one from this list)
    - Personality modes for the avatar
    - Multiple avatar selection
    - Custom backgrounds or settings
    - Special effects or animations
    - Memory/context features
    - Voice input capabilities
    - Any creative feature of their design

3. **Prepare a Demonstration**
    - Each team member should speak during the presentation
    - Show the working application with live interaction
    - Explain technical implementation details
    - Highlight their individual contributions
    - Discuss challenges and how they were solved