To successfully develop an event management app with AI integration, you'll need a mix of **technical skills** across various domains. I'll break this down into **core areas of knowledge**, covering both the **backend** (server-side) and **frontend** (client-side) development, as well as **AI technologies** that will enable you to incorporate machine learning, NLP, and other AI features.

1. Programming Languages

You'll need to be proficient in several programming languages depending on the stack you choose for both **backend** and **frontend** development:

- **Backend Development:**
  - **Python**: Highly recommended for integrating AI/ML features (e.g., using libraries like TensorFlow, PyTorch, or Scikit-learn). It's also popular for web frameworks like Django or Flask.
  - **JavaScript (Node.js)**: Useful for server-side development if you choose JavaScript for both backend and frontend.
  - **Ruby**: If you plan to use Ruby on Rails for backend development.
  - **SQL/NoSQL**: To manage databases (PostgreSQL, MySQL, or MongoDB).
- **Frontend Development:**
  - **HTML/CSS**: Essential for structuring and styling your app's web pages.
  - **JavaScript**: Necessary for building interactive frontends, especially if you're using modern JavaScript frameworks.
  - **React.js, Vue.js, or Angular**: These are modern JavaScript frameworks used for building dynamic single-page applications (SPAs).
  - **Dart (Flutter)**: If you plan to build a cross-platform mobile app (iOS and Android).
- **Mobile App Development (Optional)**:
  - **Swift** for iOS or **Kotlin/Java** for Android.
  - **React Native or Flutter**: If you want to build a cross-platform mobile app that works on both iOS and Android.

2. AI & Machine Learning

To incorporate AI in your app, you'll need to learn **Machine Learning** (ML), **Natural Language Processing** (NLP), and other AI concepts. Here's what you should focus on:

- **Machine Learning Basics**: Learn how machine learning models work, including supervised vs unsupervised learning, classification, regression, etc.
  - Learn popular algorithms like linear regression, decision trees, random forests, KNN, etc.
  - **Resources**: Coursera, edX, or Udemy have great ML courses for beginners and intermediate learners.
- **Libraries/Frameworks**:
  - **TensorFlow & Keras**: For deep learning (if you plan to use complex models like image recognition or deep neural networks).
  - **PyTorch**: Another powerful framework for deep learning.
  - **Scikit-learn**: For classical machine learning algorithms and models.
  - **Hugging Face**: For NLP tasks such as chatbots, text generation, and sentiment analysis.
- **Natural Language Processing (NLP)**:
  - Learn how to process and analyze human language using tools like **SpaCy**, **NLTK**, and **Hugging Face**.
  - **Pretrained Models**: Explore models like GPT (OpenAI), BERT, or T5 for chatbots and text generation.
- **AI in Cloud Platforms**:
  - Learn how to use AI services offered by cloud providers like Google Cloud AI, AWS AI/ML, and Azure Cognitive Services for pre-built machine learning models like language translation, image recognition, or chatbots.

3. Database Management

- **Relational Databases**: Learn how to design and query databases using **SQL** (PostgreSQL, MySQL) for storing event, ticketing, and user data.
- **NoSQL Databases**: For more flexible, scalable data storage (MongoDB, Firebase, or CouchDB).
- **Database Design & Optimization**: Learn about normalization, indexing, and query optimization.

4. Web Development Frameworks

- **Backend Frameworks**:
  - **Django** (Python): A robust web framework that is useful for building data-driven apps quickly.
  - **Flask** (Python): A lightweight framework that gives you more control over the app structure.
  - **Express.js** (Node.js): A minimal web framework for building backend services in JavaScript.
  - **Ruby on Rails** (Ruby): A powerful framework that speeds up web app development.
- **Frontend Frameworks**:
  - **React.js**: A JavaScript library for building interactive UIs.
  - **Vue.js**: Another JavaScript framework for building modern UIs, known for its simplicity and flexibility.
  - **Angular**: A comprehensive framework for building dynamic web apps, developed by Google.

5. Cloud Infrastructure & Deployment

You'll need to learn how to deploy and manage your app in the cloud for scalability and high availability:

- **Cloud Providers**:
  - **Amazon Web Services (AWS)**, **Google Cloud Platform (GCP)**, or **Microsoft Azure** for hosting, data storage, and AI services.
  - **Heroku**: A simpler platform-as-a-service (PaaS) for hosting small-to-medium applications without worrying about infrastructure management.
- **Containerization**:
  - Learn **Docker** for containerizing your app, making it easier to deploy across different environments.
  - **Kubernetes**: For orchestrating and managing containers if your app becomes large-scale.

6. Payment Integration

You'll need to learn how to integrate **payment systems** for ticket sales:

- **Stripe** or **PayPal** SDKs/APIs: For handling online payments.
- Learn about **payment security** (PCI DSS compliance) and **secure transactions**.

7. User Authentication & Authorization

- Learn how to implement **user authentication** (sign-ups, logins) using tools like **OAuth**, **JWT (JSON Web Tokens)**, or Firebase Authentication.
- Learn how to protect sensitive user data with encryption and secure authentication practices.

8. Security Best Practices

- Understand how to secure your app against common vulnerabilities (SQL injection, XSS, CSRF, etc.).
- Learn about data encryption, secure authentication methods, and user privacy laws like **GDPR**.

9. DevOps & Continuous Deployment

Learn how to manage your app’s deployment pipeline:

- **Version Control**: Use **Git** and platforms like **GitHub** or **GitLab** for source code management.
- **CI/CD (Continuous Integration/Continuous Deployment)**: Learn how to automate your testing and deployment process.

10. UX/UI Design

A simplified and intuitive user experience is essential:

- Learn the basics of **UI/UX design** using tools like **Figma**, **Sketch**, or **Adobe XD**.
- Understand design principles such as **simplicity**, **visual hierarchy**, and **responsive design**.
- If you're not a designer, consider collaborating with a professional designer to ensure your app is user-friendly.

Suggested Learning Path

1. **Learn Web Development** (HTML, CSS, JavaScript, React or Vue.js, Backend in Python/Django or Node.js).
2. **Learn the Basics of Machine Learning** (Understand algorithms and get hands-on with simple models using Scikit-learn or TensorFlow).
3. **Integrate AI Features** (Add recommendation systems, chatbots, and personalization features using ML models).
4. **Learn Database Management** (SQL, MongoDB, Firebase).
5. **Learn Cloud Services** (AWS, GCP, or Azure for deployment and AI tools).
6. **Study Mobile App Development** (Flutter or React Native if you want to go cross-platform).
7. **Build Projects**: Start building simple apps, and then gradually integrate AI features like dynamic ticket pricing or a recommendation system.
8. **Stay Updated**: AI, web development, and cloud technologies evolve quickly, so stay updated with the latest trends, tools, and best practices.