**Website Development Guide for AI First Academy**

### 1. Overview

This document serves as a complete guide to building a website for **AI First Academy**, inspired by AI First Approach. It outlines front-end and back-end functionality, technology stack, student and admin portal features, and payment gateway integration.

### 2. Technology Stack

* **Frontend:** React.js (with Tailwind CSS or Bootstrap)
* **Backend:** Python with Flask
* **Database:** PostgreSQL
* **Authentication:** JWT-based authentication
* **Payment Gateway:** Stripe or Razorpay
* **Hosting:** AWS/GCP/Netlify (frontend), Heroku/AWS EC2 (backend)

### 3. Core Functional Modules

#### 3.1. Public Website Pages

* Home Page
* Courses Listing
* Course Details Page
* About Us
* Instructors
* Admissions
* Events/Webinars
* Blogs
* FAQs
* Contact Us

#### 3.2. User Management

* **Signup/Login for Students**
  + Email/password
  + Google OAuth (optional)
* **Student Dashboard**
  + Profile update
  + Enrolled courses
  + Course progress
  + Certificates
  + Payments history

#### 3.3. Admin Portal

* Add/edit/delete courses
* Manage instructors
* Manage students
* View payments
* Upload resources
* Schedule live sessions
* Analytics Dashboard (student enrollment, payment stats)

#### 3.4. Course Management

* Add course details:
  + Title, Description
  + Curriculum (modules, lessons)
  + Video uploads
  + PDFs/Assignments
* Access control:
  + Preview (free)
  + Full access (after payment)

#### 3.5. Payment Integration

* Secure checkout using Stripe or Razorpay
* Invoice generation
* Payment confirmation email
* Refund handling (manual/admin)

#### 3.6. Live Classes Integration

* Zoom/Google Meet integration
* Calendar for upcoming sessions
* Join links in dashboard

#### 3.7. Communication Features

* Notifications (course updates, payment, events)
* Email reminders
* Chat/Discussion Forum (optional)

#### 3.8. Certification

* Auto-generate certificates after course completion
* Downloadable PDFs with verification link

#### 3.9. SEO and Performance

* Meta tags for all pages
* Sitemap and robots.txt
* Lazy loading images and videos

#### 3.10. Security Features

* HTTPS (SSL)
* Input sanitization
* Rate limiting APIs
* Encrypted passwords (bcrypt)

### 4. Milestones for Development

1. **Week 1–2: Design and Architecture**
   * Wireframes
   * Database schema
2. **Week 3–4: Frontend UI Development**
   * Static pages
   * Components for Dashboard
3. **Week 5–6: Backend APIs (Flask)**
   * Auth, Course, Payment APIs
4. **Week 7–8: Integration & Testing**
   * Connect frontend with backend
   * Test workflows and payments
5. **Week 9: Launch & Feedback Collection**

### 5. Developer Notes

* Use RESTful API structure
* Ensure mobile responsiveness
* Write clean, modular code with comments
* Maintain separate dev/staging/production environments

### 6. Suggested Enhancements (Post MVP)

* Mobile App (React Native or Flutter)
* Gamification (badges, leaderboards)
* AI-based course recommendations
* Personalized learning roadmap
* Voice Assistant Integration for navigation
* Affiliate marketing and referral programs
* Integrated chatbot for query resolution
* LMS analytics dashboard for instructors

**End of Document**