Revised 13-Day Action Plan for Complete Beginners

Overview

Students: Complete beginners (no HTML, CSS, Bootstrap, Python knowledge)
Duration: 13 days × 2 hours/day = 26 hours total
Strategy: Teach fundamentals first, then connect to project

Phase 1: Fundamentals (Days 1-6)

Day 1: Introduction to Web & HTML Basics

Hour 1: Web Fundamentals

- How websites work (Client-Server model)
- What is Frontend vs Backend?
- Introduction to HTML, CSS, JavaScript
- Understanding web browsers and developer tools

Hour 2: HTML Basics - Part 1

- HTML structure: `<!DOCTYPE>`, `<html>`, `<head>`, `<body>`
- Essential tags: headings, paragraphs, links, images
- Lists (ordered and unordered)
- **Practice:** Create a simple "About Sports Facility" page

Deliverable: A basic HTML page with text, images, and links

Day 2: HTML Forms & More Tags

Hour 1: HTML Basics - Part 2

- Divs and spans for structure
- Tables for data display
- Semantic HTML: `<header>`, `<footer>`, `<nav>`, `<section>`
- Show examples from the booking project

Hour 2: HTML Forms

- Form elements: input, textarea, select, button
- Input types: text, email, password, date, time
- Form attributes: action, method, name
- **Practice:** Create a "Book a Turf" form (without styling)

Deliverable: Registration and booking form in plain HTML

Day 3: CSS Fundamentals

Hour 1: CSS Basics

- What is CSS and why we need it?
- Three ways to add CSS (inline, internal, external)
- Selectors: element, class, id
- Colors, fonts, text styling
- **Practice:** Style the forms created yesterday

Hour 2: CSS Box Model & Layout

- Padding, margin, border
- Width and height
- Display property (block, inline, inline-block)
- Basic positioning
- **Practice:** Create a card layout for sports facilities

Deliverable: Styled booking form with proper spacing and colors

---

Day 4: Bootstrap Framework

Hour 1: Introduction to Bootstrap

- What is Bootstrap and why use it?
- Adding Bootstrap to HTML (CDN)
- Grid system basics (container, row, col)
- Responsive design concept
- Show project's Bootstrap usage

Hour 2: Bootstrap Components

- Buttons and button groups
- Cards for facility display
- Forms and form controls
- Navbar basics
- **Practice:** Recreate booking form using Bootstrap

Deliverable: Responsive booking page using Bootstrap

Day 5: Python Basics - Part 1

Hour 1: Python Introduction

- Why Python for web development?
- Installing Python and running first program
- Variables and data types (strings, integers, floats, booleans)
- Basic operators and expressions
- Print statements and comments

Hour 2: Python Control Flow

- If-else statements
- Comparison operators
- Logical operators (and, or, not)
- **Practice:** Program to check if a time slot is available
- **Example:** "If booking_time is between 9 AM and 10 PM, accept booking"

Deliverable: 3-4 simple Python programs with conditions

Day 6: Python Basics - Part 2

Hour 1: Lists and Loops

- Lists and list operations
- For loops to iterate through facilities
- While loops
- **Practice:** Create a list of sports facilities and print them

Hour 2: Functions and Dictionaries

- Creating functions with def
- Parameters and return values
- Dictionaries (key-value pairs)
- **Practice:** Function to calculate booking price
- **Example:** Dictionary to store facility details

Deliverable: Python functions for booking calculations

Phase 2: Connecting to Project (Days 7-11)

Day 7: Introduction to Django

Hour 1: Django Basics

- What is Django framework?
- MVC/MVT architecture explained simply
- Installing Django and creating first project
- Project structure walkthrough
- Understanding manage.py

Hour 2: Setting Up the Booking Project

- Download and extract the project
- Create virtual environment
- Install requirements.txt
- Run migrations
- Access admin panel
- **Show:** How their HTML/CSS connects to Django templates

Deliverable: Project running on localhost:8000

Day 8: Django Models & Database

Hour 1: Understanding Models

- What is a database (simple explanation)
- Django models as Python classes
- **Examine project models:**
  - User model (name, email, password)
  - Facility model (name, location, price)
  - Booking model (user, facility, date, time)
- Field types: CharField, IntegerField, DateField

Hour 2: Django Admin & Data Entry

- Creating superuser
- Registering models in admin
- Adding sample facilities through admin
- Adding sample users
- **Practice:** Add 5 facilities, 3 users, 2 bookings

Deliverable: Database populated with sample data

Day 9: Django Views & URLs

Hour 1: URLs and Routing

- How URLs work in Django
- urls.py file structure
- URL patterns
- **Examine project URLs:**
  - Homepage URL
  - Booking URL
  - Login/Register URLs
- Connecting URLs to views

Hour 2: Views - The Logic Layer

- What are views?
- Function-based views in the project
- **Examine key views:**
  - Facility listing view (fetches all facilities)
  - Booking creation view (creates booking)
- How views pass data to templates

Deliverable: Understanding of 3-4 main project URLs and views

---

Day 10: Django Templates

Hour 1: Template Basics

- Template language syntax: `{{ variable }}`, `{% tag %}`
- Template inheritance (base.html → child templates)
- **Examine project templates:**
  - base.html (common header/footer)
  - home.html (facility listing)
  - booking_form.html

Hour 2: Template Customization

- Displaying data from views: `{{ facility.name }}`
- Loops in templates: `{% for facility in facilities %}`
- If conditions in templates: `{% if user.is_authenticated %}`
- **Practice:** Modify homepage to add a welcome message
- Change a color or heading

Deliverable: Customized homepage with student's changes

Day 11: Forms & User Input

Hour 1: Django Forms

- Understanding forms.py
- **Examine project forms:**
  - Registration form
  - Booking form
- Form validation basics
- How forms connect to models

Hour 2: Form Processing

- GET vs POST requests (simple explanation)
- How form data is saved to database
- Success/error messages
- **Trace the booking flow:**
  1. User fills form
  2. Data goes to view
  3. View saves to database
  4. Confirmation page shown

Deliverable: Create a test booking from form to database

Phase 3: Integration & Testing (Days 12-13)

Day 12: JavaScript in the Project

Hour 1: JavaScript Basics & Project Usage

- What JavaScript does (client-side interactivity)
- Variables, functions, events (very basic)
- **Examine project JavaScript:**
  - Date picker functionality
  - Real-time availability checking
  - Form validation
- Where JavaScript files are located

Hour 2: Making Small Modifications

- Modify an alert message
- Change date picker settings
- Add a simple button click event
- **Practice:** Add confirmation dialog before booking
- **Example:** "Are you sure you want to book this slot?"

Deliverable: One working JavaScript modification

Day 13: Complete Flow Testing & Review

Hour 1: End-to-End Testing

- **Test complete user journey:**
  1. Register a new user
  2. Login
  3. Browse facilities
  4. Make a booking
  5. View booking history
  6. Admin reviews booking
- Document the flow with screenshots
- Identify each file involved at each step

Hour 2: Q&A and Next Steps

- Review all concepts learned
- Clarify doubts
- **Small challenge:** "Change the color scheme of one page"
- Discussion on how to continue learning
- Resources for further practice

Deliverable: Complete project demonstration by students

Daily Structure Template

```
Every Session:
├─ 0-5 min: Quick recap & doubt clearing
├─ 5-50 min: New concept teaching (hands-on)
├─ 50-110 min: Guided practice
└─ 110-120 min: Summary & homework assignment
```

Teaching Resources Needed

Visual Aids:

- Flowcharts showing how web requests work
- Diagrams of Django MVT architecture
- Screenshots of project at each stage

Code Samples:

- Simple HTML templates for practice
- Python programs starting from "Hello World"
- Commented code snippets from the project

Tools:

- VS Code with extensions
- Browser Developer Tools (F12)
- DB Browser for SQLite (to visualize database)

Homework Assignments (15-20 min each day)

- **Day 1-2:** Create HTML pages for different sections
- **Day 3-4:** Style existing HTML, experiment with Bootstrap
- **Day 5-6:** Write 3-5 Python programs
- **Day 7-8:** Explore Django admin, add more data
- **Day 9-10:** Read and understand one view function
- **Day 11-12:** Trace one complete feature from URL to template
- **Day 13:** Prepare questions and improvements

Key Teaching Principles

1. **Show, Don't Just Tell:** Live code everything
2. **Connect to Project:** After teaching each concept, show where it's used in the booking system
3. **Incremental Complexity:** Start with "Hello World", end with understanding the project
4. **Hands-On Practice:** Students must type code, not just watch
5. **Immediate Application:** Each concept learned should be immediately applied

---

Success Metrics

By Day 13, students should be able to:

- ✅ Create a basic HTML page with forms
- ✅ Style elements using CSS and Bootstrap
- ✅ Write simple Python programs with functions
- ✅ Understand Django project structure
- ✅ Navigate the codebase confidently
- ✅ Make small modifications (text, colors, simple logic)
- ✅ Explain the booking flow from user action to database

Backup Plan (If Students Struggle)

- **Reduce scope:** Focus more on HTML/CSS/Python basics, less on Django internals
- **Extend practice:** Give same exercises twice with variations
- **Pair programming:** Students work in pairs
- **Recorded sessions:** Provide recordings for review

Additional Resources for Students

Online Learning Platforms:

1. **HTML/CSS:**
   - W3Schools (w3schools.com)
   - MDN Web Docs (developer.mozilla.org)
2. **Python:**
   - Python.org official tutorial
   - Real Python (realpython.com)
3. **Django:**
   - Django official documentation (djangoproject.com)
   - Django Girls Tutorial
4. **Bootstrap:**
   - Bootstrap official documentation (getbootstrap.com)

Practice Websites:

- Codecademy
- FreeCodeCamp
- Coursera (Python for Everybody)

Notes for Instructor

Preparation Checklist:

- [ ] Install all required software on lab computers
- [ ] Prepare sample code files for each day
- [ ] Create a shared folder for code distribution
- [ ] Prepare troubleshooting guide for common errors
- [ ] Have project backup ready
- [ ] Prepare assessment questions

Common Challenges to Expect:

1. **Installation Issues:** Be ready with offline installers
2. **Syntax Errors:** Emphasize importance of indentation in Python
3. **Path Issues:** Teach basic command line navigation
4. **Port Conflicts:** Know how to change Django development server port
5. **Database Errors:** Keep a fresh copy of migrations

Assessment Strategy:

- **Informal:** Daily check of deliverables
- **Mid-point (Day 7):** Quick quiz on HTML/CSS/Python basics
- **Final (Day 13):** Students demonstrate one feature modification

---

Project Modification Ideas for Practice

Easy Modifications:

1. Change website colors and logo
2. Add a new field to booking form (e.g., "Special Requirements")
3. Modify welcome message on homepage
4. Change number of facilities displayed per page

Medium Modifications:

1. Add a new facility type/category
2. Create an "About Us" page
3. Modify booking confirmation email template
4. Add facility image upload

Challenging Extensions (Post-Training):

1. Add user reviews/ratings for facilities
2. Implement booking cancellation feature
3. Add SMS notifications
4. Create advanced search filters

Contact & Support Information

For Technical Doubts:

- Create a WhatsApp/Telegram group
- Schedule office hours (30 min after each session)
- Maintain a shared Google Doc for FAQ

For Project Files:

- GitHub repository link
- Google Drive backup
- USB drive backup for offline access

Certificate/Completion Criteria

Students will receive completion recognition upon:

1. ✅ Attendance of at least 11 out of 13 sessions
2. ✅ Submission of 80% of homework assignments
3. ✅ Successful demonstration of final project understanding
4. ✅ Completion of one custom modification

Last Updated: November 2025
Prepared for: Sports Turf Playground Booking System Training
Duration: 13 Days (26 Hours Total)