

```

import os
import csv
from fpdf import FPDF

# --- Configuration ---
OUTPUT_DIR = "HRbiz_Forms_Proocols_Logs"
COMPANY_NAME = "Button Nose Grooming"
CONSULTANT = "HRbiz.org"
NAVY = (26, 60, 94)
TEAL = (46, 139, 153)
GREY = (200, 200, 200)

if not os.path.exists(OUTPUT_DIR):
    os.makedirs(OUTPUT_DIR)

class BrandedPDF(FPDF):
    def header(self):
        self.set_font('Arial', 'B', 12)
        self.set_text_color(*NAVY)
        self.cell(0, 10, 'HRbiz.org | Compliance System', 0, 1, 'R')
        self.ln(5)

    def footer(self):
        self.set_y(-15)
        self.set_font('Arial', 'I', 8)
        self.set_text_color(128)
        self.cell(0, 10, f'Page {self.page_no()} - {COMPANY_NAME}\nInternal Doc', 0, 0, 'C')

    def create_form_title(self, title):
        self.set_font('Arial', 'B', 16)
        self.set_text_color(*NAVY)
        self.cell(0, 10, title, 0, 1, 'L')
        self.set_draw_color(*TEAL)
        self.set_line_width(1)
        self.line(10, self.get_y(), 200, self.get_y())
        self.ln(10)

    def add_section_header(self, text):
        self.ln(5)
        self.set_font('Arial', 'B', 11)
        self.set_fill_color(240, 240, 240)
        self.set_text_color(*NAVY)
        self.cell(0, 8, text, 0, 1, 'L', fill=True)
        self.ln(2)

    def add_field(self, label, lines=1):
        self.set_font('Arial', '', 10)

```

```

        self.set_text_color(0)
        self.cell(0, 8, f"{label}:", 0, 1)
        self.set_draw_color(*GREY)
        self.set_line_width(0.2)
        for _ in range(lines):
            self.cell(0, 8, "", "B", 1) # Bottom border only creates a
writing line
        self.ln(2)

def add_checkbox(self, label):
    self.set_font('Arial', '', 10)
    self.cell(5, 5, "", 1, 0) # Box
    self.cell(2, 5, "", 0, 0) # Spacer
    self.cell(0, 5, label, 0, 1)
    self.ln(2)

def add_text_block(self, text):
    self.set_font('Arial', '', 10)
    self.multi_cell(0, 5, text)
    self.ln(5)

# --- Content Definitions ---

def generate_complaint_forms():
    # 1. Employee Complaint Form
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Employee Complaint Form")
    pdf.add_text_block("Confidentiality: This form will be handled
with the highest degree of confidentiality possible. Retaliation
against anyone filing a complaint is strictly prohibited.")
    pdf.add_section_header("Reporter Information")
    pdf.add_field("Name")
    pdf.add_field("Job Title")
    pdf.add_field("Date of Report")
    pdf.add_section_header("Incident Details")
    pdf.add_field("Date/Time of Incident")
    pdf.add_field("Location")
    pdf.add_field("Name(s) of Person(s) Involved")
    pdf.add_field("Witnesses (if any)")
    pdf.add_field("Description of Incident (Please be specific)",
lines=6)
    pdf.add_field("Desired Outcome", lines=3)
    pdf.add_section_header("Acknowledgment")
    pdf.add_text_block("I certify that the information provided is
true and correct to the best of my knowledge.")
    pdf.add_field("Signature")
    pdf.add_field("Date")

```

```

pdf.output(f"{OUTPUT_DIR}/complaint_form_employee.pdf")

# 2. Third Party Complaint Form
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Third-Party Complaint Form")
pdf.add_text_block("For use by Customers, Vendors, or
Volunteers.")
pdf.add_section_header("Complainant Information")
pdf.add_field("Name")
pdf.add_field("Relationship to Business (e.g., Customer)")
pdf.add_field("Contact Phone/Email")
pdf.add_section_header("Incident Details")
pdf.add_field("Date/Time")
pdf.add_field("Description of Incident", lines=6)
pdf.output(f"{OUTPUT_DIR}/complaint_form_third_party.pdf")

def generate_investigation_docs():
    # 3. Witness Statement
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Witness Statement Form")
    pdf.add_field("Witness Name")
    pdf.add_field("Interviewer Name")
    pdf.add_section_header("Statement")
    pdf.add_text_block("Please describe exactly what you saw or heard.
Distinguish between facts (what you saw) and opinions (what you
think).")
    pdf.add_field("Narrative", lines=10)
    pdf.add_field("Signature")
    pdf.output(f"{OUTPUT_DIR}/witness_statement_form.pdf")

    # 4. Incident Report (Safety/General)
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Workplace Incident Report")
    pdf.add_section_header("Incident Type")
    pdf.add_checkbox("Safety / Injury")
    pdf.add_checkbox("Animal Handling / Bite")
    pdf.add_checkbox("Property Damage")
    pdf.add_checkbox("Customer Misconduct")
    pdf.add_section_header("Details")
    pdf.add_field("Description", lines=5)
    pdf.add_field("Action Taken", lines=3)
    pdf.output(f"{OUTPUT_DIR}/incident_report_form.pdf")

    # 5. Investigation Intake (Internal)
    pdf = BrandedPDF()

```

```

pdf.add_page()
pdf.create_form_title("Investigation Intake Form (Internal Use)")
pdf.add_section_header("Triage Assessment")
pdf.add_checkbox("Does this allege harassment based on a protected
class?")
pdf.add_checkbox("Is there an immediate safety risk?")
pdf.add_checkbox("Does this require an external PI/Attorney?
(Serious allegations)")
pdf.add_field("Allegations Summary", lines=4)
pdf.add_field("Assigned To")
pdf.output(f"{OUTPUT_DIR}/investigation_intake_form.pdf")

# 6. Case File Template
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Investigation Case File Cover Sheet")
pdf.add_field("Case Number")
pdf.add_field("Complainant")
pdf.add_field("Respondent (Accused)")
pdf.add_section_header("Chronology Log")
pdf.add_field("Date / Action", lines=8)
pdf.add_section_header("Evidence Log")
pdf.add_field("Item / Location", lines=5)
pdf.output(f"{OUTPUT_DIR}/investigation_case_file_template.pdf")

# 7. Closure Summary
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Investigation Closure Summary")
pdf.add_section_header("Findings")
pdf.add_field("Summary of Facts", lines=5)
pdf.add_section_header("Conclusion")
pdf.add_checkbox("Sustained (Policy Violation Found)")
pdf.add_checkbox("Not Sustained (Insufficient Evidence)")
pdf.add_checkbox("Unfounded (Did not occur)")
pdf.add_field("Corrective Action Taken", lines=3)
pdf.output(f"{OUTPUT_DIR}/investigation_closure_summary.pdf")

def generate_corrective_action():

# 8. Corrective Action Form
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Corrective Action Notice")
pdf.add_field("Employee Name")
pdf.add_field("Date")
pdf.add_section_header("Action Level")
pdf.add_checkbox("Verbal Warning (Documented)")
pdf.add_checkbox("Written Warning")

```

```

pdf.add_checkbox("Final Warning")
pdf.add_checkbox("Termination")
pdf.add_section_header("Policy Violation")
pdf.add_field("Description of conduct", lines=5)
pdf.add_section_header("Improvement Plan")
pdf.add_field("Required changes", lines=4)
pdf.add_field("Employee Comments", lines=3)
pdf.add_field("Employee Signature")
pdf.add_field("Manager Signature")
pdf.output(f"{OUTPUT_DIR}/corrective_action_form.pdf")

def generate_training_docs():
    # 9. Training Certs (Employee)
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.set_font('Arial', 'B', 24)
    pdf.set_text_color(*NAVY)
    pdf.cell(0, 20, "Certificate of Completion", 0, 1, 'C')
    pdf.ln(10)
    pdf.set_font('Arial', '', 14)
    pdf.set_text_color(0)
    pdf.cell(0, 10, "This certifies that:", 0, 1, 'C')
    pdf.add_field("Employee Name")
    pdf.ln(5)
    pdf.cell(0, 10, "Has successfully completed:", 0, 1, 'C')
    pdf.set_font('Arial', 'B', 16)
    pdf.cell(0, 10, "CA Harassment Prevention Training (1 Hour)", 0,
1, 'C')
    pdf.ln(10)
    pdf.set_font('Arial', '', 10)
    pdf.add_field("Date Completed")
    pdf.add_field("Trainer / Provider (HRbiz.org)")

pdf.output(f"{OUTPUT_DIR}/employee_training_certificate_template.pdf")

# 10. Training Certs (Supervisor)
pdf = BrandedPDF()
pdf.add_page()
pdf.set_font('Arial', 'B', 24)
pdf.set_text_color(*NAVY)
pdf.cell(0, 20, "Certificate of Completion", 0, 1, 'C')
pdf.set_font('Arial', '', 14)
pdf.set_text_color(0)
pdf.cell(0, 10, "This certifies that:", 0, 1, 'C')
pdf.add_field("Supervisor Name")
pdf.ln(5)
pdf.cell(0, 10, "Has completed the mandated:", 0, 1, 'C')
pdf.set_font('Arial', 'B', 16)

```

```

pdf.cell(0, 10, "CA Supervisor Harassment Training (2 Hours)", 0,
1, 'C')
pdf.ln(10)
pdf.set_font('Arial', '', 10)
pdf.add_field("Date Completed")

pdf.output(f"{OUTPUT_DIR}/supervisor_training_certificate_template.pdf")

def generate_onboarding_and_ack():
    # 11. Policy Acknowledgment
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Policy Acknowledgment Form")
    pdf.add_text_block("I acknowledge receipt of the following Button
Nose Grooming policies:")
    pdf.add_checkbox("Employee Handbook / Policy Packet")
    pdf.add_checkbox("Harassment Prevention Policy")
    pdf.add_checkbox("Anti-Retaliation Policy")
    pdf.add_checkbox("Safety Protocols")
    pdf.add_text_block("I understand my employment is At-Will and
these policies do not constitute a contract.")
    pdf.add_field("Signature")
    pdf.add_field("Date")

pdf.output(f"{OUTPUT_DIR}/employee_policy_acknowledgment_form.pdf")

# 12. Supervisor Designation
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Supervisor Designation Form")
pdf.add_text_block("Under CA FEHA, any individual with authority
to direct the work of others is a supervisor.")
pdf.add_field("Employee Name")
pdf.add_section_header("Authority Review")
pdf.add_checkbox("Has authority to assign work/schedules?")
pdf.add_checkbox("Has authority to check work quality?")
pdf.add_checkbox("Has authority to correct/discipline?")
pdf.add_text_block("Based on the above, this employee is
DESIGNATED AS A SUPERVISOR and requires 2-hour training.")
pdf.add_field("Owner Signature")
pdf.output(f"{OUTPUT_DIR}/supervisor_designation_form.pdf")

# 13. Onboarding Checklist
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("CA New Hire Checklist")
pdf.add_checkbox("Form I-9 (Employment Eligibility)")

```

```

pdf.add_checkbox("Form W-4 (Federal Tax)")
pdf.add_checkbox("Form DE-4 (CA Tax)")
pdf.add_checkbox("Wage Theft Prevention Notice (2810.5)")
pdf.add_checkbox("Workers' Comp Pamphlet")
pdf.add_checkbox("DFEH-185 Brochure (Sexual Harassment)")
pdf.add_checkbox("Policy Acknowledgments Signed")
pdf.add_checkbox("Harassment Training Scheduled")
pdf.output(f"{OUTPUT_DIR}/onboarding_forms_checklist.pdf")

# 14. Interview Form
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Structured Interview Form")
pdf.add_text_block("Reminder: Do not ask about age, marital status, children, religion, or disability.")
pdf.add_field("Candidate Name")
pdf.add_field("Position")
pdf.add_field("Skill: Dog Handling Experience", lines=2)
pdf.add_field("Skill: Customer Service", lines=2)
pdf.add_field("Skill: Reliability/Schedule", lines=2)
pdf.add_field("Interviewer Notes", lines=5)
pdf.output(f"{OUTPUT_DIR}/interview_form_new_hire.pdf")

# 15. Separation Checklist
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Separation/Termination Checklist")
pdf.add_checkbox("Final Paycheck (Provided on last day?)")
pdf.add_checkbox("EDD Unemployment Booklet Given")
pdf.add_checkbox("COBRA / Cal-COBRA Notice")
pdf.add_checkbox("Return of Keys/Equipment")
pdf.add_checkbox("Computer Access Revoked")
pdf.output(f"{OUTPUT_DIR}/separation_checklist.pdf")

def generate_accommodation_docs():
    # 16. Medical Accommodation Request
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Reasonable Accommodation Request")
    pdf.add_field("Employee Name")
    pdf.add_field("Nature of limitation (Do not disclose diagnosis)", lines=3)
    pdf.add_field("Requested Accommodation", lines=3)
    pdf.add_field("Duration Needed")
    pdf.output(f"{OUTPUT_DIR}/medical_accommodation_request_form.pdf")

    # 17. Pregnancy Accommodation
    pdf = BrandedPDF()

```

```

pdf.add_page()
pdf.create_form_title("Pregnancy Accommodation Request")
pdf.add_field("Employee Name")
pdf.add_text_block("CA Law provides specific rights for pregnancy, childbirth, and related conditions.")
pdf.add_field("Requested Change (e.g. Chair, Breaks)", lines=3)

pdf.output(f"{OUTPUT_DIR}/pregnancy_accommodation_request_form.pdf")

# 18. Interactive Process Doc
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Interactive Process Record")
pdf.add_text_block("Use this to document the 'good faith' meeting regarding accommodations.")
pdf.add_field("Date of Meeting")
pdf.add_field("Attendees")
pdf.add_field("Options Discussed", lines=5)
pdf.add_field("Final Decision / Plan", lines=3)

pdf.output(f"{OUTPUT_DIR}/interactive_process_documentation_form.pdf")

def generate_protocols():
    # 19. Retaliation Protocol
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Retaliation Prevention Protocol")
    pdf.add_text_block("1. Upon receipt of any complaint, remind all supervisors strictly: NO adverse action.\n2. Monitor the complainant's schedule and workload weekly.\n3. Check in with complainant: 'How are things going?'")
    pdf.output(f"{OUTPUT_DIR}/retaliation_prevention_protocol.pdf")

    # 20. Complaint Intake Protocol
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Complaint Intake Protocol")
    pdf.add_text_block("1. Listen without judgment.\n2. Do not promise absolute confidentiality.\n3. Document everything immediately.\n4. Determine if safety is at risk.")
    pdf.output(f"{OUTPUT_DIR}/complaint_intake_protocol.pdf")

    # 21. Chain of Custody
    pdf = BrandedPDF()
    pdf.add_page()
    pdf.create_form_title("Evidence Chain of Custody")
    pdf.add_field("Item Description")
    pdf.add_field("Collected By")

```

```
pdf.add_field("Date/Time")
pdf.add_field("Storage Location")

pdf.output(f"{OUTPUT_DIR}/documentation_chain_of_custody_form.pdf")

# 22. Document Retention
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Document Retention Policy")
pdf.add_text_block("Keep for at least 3 years:\n- Payroll
Records\n- Hiring Docs\n\nKeep for at least 2 years:\n- Harassment
Training Records\n- Personnel Files\n\nStore in locked cabinet or
encrypted drive.")
pdf.output(f"{OUTPUT_DIR}/document_retention_policy.pdf")

# 23. Online Training Instructions
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Online Training Instructions")
pdf.add_text_block("To access your Harassment Prevention
Training:\n1. Check your email for an invite from HRbiz.org.\n2.
Click the link to register.\n3. Complete the interactive video modules
(1 hr for staff, 2 hr for leads).\n4. Download your certificate upon
completion.\n5. Email certificate to Judy.")

pdf.output(f"{OUTPUT_DIR}/online_training_access_instructions.pdf")

# 24. FEHA Reasonable Accommodation Protocol
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("FEHA Accommodation Protocol")
pdf.add_text_block("1. Employee requests help.\n2. Engage in
'Interactive Process' (Meeting).\n3. Review medical restrictions (if
applicable).\n4. Determine if reasonable accommodation exists.\n5.
Document the outcome.")

pdf.output(f"{OUTPUT_DIR}/FEHA_reasonable_accommodation_protocol.pdf")

# 25. Investigation SOP
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Investigation SOP")
pdf.add_text_block("Standard Operating Procedure for Internal
Reviews:\n1. Receive Complaint.\n2. Triage (Internal vs External
PI).\n3. Interview Complainant.\n4. Interview Witnesses.\n5. Interview
Accused.\n6. Make Finding.\n7. Close Case.")

pdf.output(f"{OUTPUT_DIR}/investigation_standard_operating_procedure.p
```

```

df")

# 26. Safety Incident Form
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("General Safety Incident Form")
pdf.add_field("Date/Time")
pdf.add_field("Type of Injury")
pdf.add_field("Description", lines=5)
pdf.output(f"{OUTPUT_DIR}/safety_incident_form.pdf")

# 27. Animal Handling Incident
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Animal Handling Incident Report")
pdf.add_field("Pet Name / Breed")
pdf.add_field("Owner Name")
pdf.add_field("Groomer Involved")
pdf.add_field("Bite or Scratch?", lines=1)
pdf.add_field("Behavioral Trigger", lines=3)
pdf.output(f"{OUTPUT_DIR}/animal_handling_incident_report.pdf")

# 28. Customer Misconduct
pdf = BrandedPDF()
pdf.add_page()
pdf.create_form_title("Customer Misconduct Report")
pdf.add_text_block("Use this to report harassment or abuse by a client.")
pdf.add_field("Customer Name")
pdf.add_field("Incident Details", lines=5)
pdf.output(f"{OUTPUT_DIR}/customer_misconduct_report_form.pdf")

def generate_logs():
    # Generating CSV files for logs
    # 29. Training Roster
    with open(f"{OUTPUT_DIR}/training_roster_template.csv", 'w',
newline='') as f:
        writer = csv.writer(f)
        writer.writerow(["Employee Name", "Role", "Supervisor? (Y/N)", "Training Type", "Date Completed", "Cert on File?", "Next Due Date"])
        writer.writerow(["Sample: Jane Doe", "Bather", "N", "1-Hour Employee", "2024-01-15", "Y", "2026-01-15"])

    # 30. Biennial Schedule
    with open(f"{OUTPUT_DIR}/biennial_training_schedule.csv", 'w',
newline='') as f:
        writer = csv.writer(f)

```

```

        writer.writerow(["Year", "Training Cycle", "Judy (Owner)", "Manager A", "Groomer B", "Bather C"])
        writer.writerow(["2024", "Due by Jan 31", "Completed", "Completed", "Pending", "Completed"])
        writer.writerow(["2026", "Due by Jan 31", "Due", "Due", "Due", "Due"])

# 31. Policy Distribution Log
with open(f"{OUTPUT_DIR}/policy_distribution_log.csv", 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(["Employee Name", "Policy Version", "Date Given", "Signature Received? (Y/N)", "Notes"])
    writer.writerow(["John Smith", "v2024.1", "2024-01-10", "Y", "Orientation"])

# 32. Onboarding Completion Log
with open(f"{OUTPUT_DIR}/onboarding_completion_log.csv", 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(["New Hire Name", "Start Date", "I-9 Complete", "Handbook Signed", "Harassment Training Sent", "Training Done"])
    writer.writerow(["Alice New", "2024-02-01", "Y", "Y", "Y", "Pending"])

if __name__ == "__main__":
    print(f"Generating HR Compliance Assets for {COMPANY_NAME}...")
    generate_complaint_forms()
    generate_investigation_docs()
    generate_corrective_action()
    generate_training_docs()
    generate_onboarding_and_ack()
    generate_accommodation_docs()
    generate_protocols()
    generate_logs()
    print(f"Success! All files located in {OUTPUT_DIR}/")

```

Deployment Instructions

- Save the Script:** Copy the code block above into a file named `generate_compliance_forms.py`.
- Run the Script:**

```
python generate_compliance_forms.py
```