

2. Free IT Career PDF Toolkit

Filename: IT_Career_Toolkit.zip

Contents:

1. Project Documentation Template: project_template.docx (and PDF version).
2. Certification Tracker: certification_tracker.xlsx.
3. Automation Scripts:
 - o merge_certifications.py
 - o add_metadata.py
 - o update_resume.py
4. README.md (Instructions).

project_template.docx/PDF

Structure:

- Project Title: CI/CD Pipeline Automation
- Problem Statement: Manual deployments took 2+ hours.
- Solution: Built Jenkins pipelines with Python scripting.
- Outcome: Reduced deployment time to 15 minutes.
- Code Snippets:

python Copy

Sample Jenkinsfile

```
pipeline {
  agent any
  stages {
    stage('Build') { steps { sh 'python build.py' }}
    stage('Deploy') { steps { sh 'docker-compose up -d' }}
  }
}
```

merge_certifications.py

python Copy

```
from PyPDF2 import PdfMerger
import os
```

```
def merge_certificates(input_folder, output_file):
    merger = PdfMerger()
    for filename in sorted(os.listdir(input_folder)):
        if filename.endswith(".pdf"):
            merger.append(os.path.join(input_folder, filename))
    merger.write(output_file)
```

```
merge_certificates("./certs", "certification_portfolio.pdf")
```



add_metadata.py

python Copy

```
from PyPDF2 import PdfReader, PdfWriter
```

```
def tag_pdf(input_pdf, output_pdf, metadata):
    reader = PdfReader(input_pdf)
    writer = PdfWriter()
    writer.append(reader)
    writer.add_metadata(metadata)
    with open(output_pdf, "wb") as f:
        writer.write(f)
```

```
metadata = {
    "/Title": "AWS Certified Solutions Architect",
    "/Author": "John Doe",
    "/Subject": "AWS Certification 2023"
}
tag_pdf("aws_cert.pdf", "aws_cert_tagged.pdf", metadata)
```

update_resume.py

python Copy

```
import pdfrw
```

```
from datetime import datetime
```

```
def update_experience(template_path, output_path, new_role):
    template = pdfrw.PdfReader(template_path)
    annotations = template.pages[0]['/Annots']
    for annotation in annotations:
        if annotation['/T'] == 'Experience':
            new_text = f"{new_role['title']} | {new_role['company']} ({datetime.now().year})"
            annotation.update(pdfrw.PdfDict(V=new_text))
    pdfrw.PdfWriter().write(output_path, template)
```

```
new_job = {"title": "Cloud Architect", "company": "Tech Corp"}
update_experience("resume_template.pdf", "updated_resume.pdf", new_job)
```

README.md for Both Resources

markdown Copy

IT Career PDF Toolkit

1. ATS Resume Template

- Use `resume\_template.tex` to create ATS-friendly resumes.
- Validate with `check\_ats\_compliance.py`.

2. Project & Certification Tools

- ****Merge Certificates****: Run `python merge\_certifications.py`.
- ****Add Metadata****: Update `metadata` in `add\_metadata.py`.
- ****Auto-Update Resumes****: Modify `new\_job` in `update\_resume.py`.

