Folder Structure  
phishshield/

├── manage.py

├── phishshield/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ └── wsgi.py

│

├── scanner/

│ ├── \_\_init\_\_.py

│ ├── admin.py ← (Unused)

│ ├── apps.py

│ ├── models.py ← For saving reports

│ ├── views.py

│ ├── urls.py

│ ├── forms.py

│ └── utils.py ← (Q-learning model logic)

│

├── templates/

│ ├── base.html

│ ├── index.html ← Home input page

│ ├── result.html ← Scan result page

│ └── report.html ← PDF/report template

│

├── static/

│ └── css/

│ └── styles.css

│

├── requirements.txt  
  
Prompt  
  
Build a Django web application named phishshield with a single app called scanner.

The system should:

Accept a URL from the user on the homepage.

Use a pre-trained Q-learning model (available in scanner/utils.py) to analyze the URL and return phishing detection results.

Display the result on a results page.

Show extracted features and confidence score.

Save the report to the database (model name: ScanReport) with fields: url, result, confidence, features (JSON), and timestamp.

Allow the user to download a simple PDF version of the report.

Do not include any login, registration, or admin pages.

Folder & File Requirements:

templates/: base.html, index.html, result.html, report.html

static/css/: styles.css

scanner/forms.py: a form with a single URL input field

scanner/models.py: ScanReport model with JSON field for features

scanner/views.py:

home\_view → displays index.html

analyze\_url\_view → handles POST, runs model, shows result.html

report\_pdf\_view → renders report.html to PDF and returns it

scanner/utils.py: a predict\_phishing(url) function that:

extracts features

runs them through the pre-trained Q-learning model

returns prediction, confidence, and features as a dictionary

scanner/urls.py: routes for /, /result/, /report/<id>/pdf/

Design:

Use base.html for layout

Use styles.css for minimal, clean, professional styling

index.html has a form to input URL

result.html displays results in bullet format with color highlights

report.html is styled for printing/PDF with no buttons or nav

Use SQLite as the database. Add basic CSRF protection. No user authentication needed. Make sure all templates extend from base.html.