**https://llamacoder.together.ai/**

**Project Structure**

cybersecurity-training-platform/

├── cybersecurity\_platform.py # Main application

├── setup.py # Packaging configuration

├── requirements.txt # Dependencies

├── README.md # Project documentation

├── build.sh # Unix build script

├── build.bat # Windows build script

├── templates/ # HTML templates

│ ├── base.html

│ ├── dashboard.html

│ ├── modules.html

│ ├── users.html

│ └── reports.html

└── static/ # Static assets (CSS, JS, images)

**Step 1: Initialize Git Repository Locally**

**V4**

Component v4

component-v4.sh

**Step 2: Connect to GitHub Repository**

# Add remote origin (replace with your actual repository URL)

git remote add origin https://github.com/jubinbalachandran/cybersecurity-training-platform.git

# Verify remote

git remote -v

# Push to GitHub

git branch -M main

git push -u origin main

**Step 3: Create Project Structure**

Create the following directory structure in your project folder:

Cybersecurity Awareness/

├── cybersecurity\_platform.py

├── requirements.txt

├── README.md

├── setup.py

├── templates/

│ ├── base.html

│ ├── dashboard.html

│ ├── modules.html

│ ├── users.html

│ └── reports.html

├── static/

│ └── (CSS, JS, images will go here)

└── docs/

└── setup\_guide.md

**Step 4: Create HTML Templates**

Create the **templates** folder and add all HTML files as shown in the previous response.

**Step 5: Create Setup Configuration**

Create **setup.py** with the content from the previous response.

**Step 6: Create Documentation**

Create **docs/setup\_guide.md** with this document's content.

**Step 7: Test the Application**

# Run the application

python cybersecurity\_platform.py

**Step 8: Access the Platform**

Open your browser and go to **http://localhost:5000**

**Step 9: Package the Application**

# Install build tools

pip install build

# Create distribution package

python -m build

**Detailed Documentation**

**Project Structure Explanation**

1. **cybersecurity\_platform.py**: Main application file with Flask routes and database logic
2. **templates/**: HTML files for the web interface
3. **static/**: CSS, JavaScript, and image files
4. **setup.py**: Configuration for packaging the application
5. **requirements.txt**: List of Python dependencies
6. **docs/**: Documentation files

**Database Schema**

The application uses SQLite with these tables:

* **modules**: Training modules information
* **users**: User accounts and risk levels
* **user\_progress**: User training completion status
* **reports**: Generated security reports

**API Endpoints**

* **/api/modules**: Returns all training modules
* **/api/users**: Returns all users
* **/api/stats**: Returns dashboard statistics

**Customization Options**

1. Add new training modules to the database
2. Modify HTML templates for branding
3. Extend API endpoints for additional features
4. Add authentication for production deployment

**Deployment Considerations**

1. For production, add authentication and authorization
2. Use a production WSGI server like Gunicorn
3. Implement HTTPS for secure communication
4. Set up regular database backups
5. Monitor application logs for security events

**Troubleshooting Common Issues**

1. **Port already in use**: Change port in **cybersecurity\_platform.py**
2. **Database errors**: Check file permissions for the database file
3. **Template not found**: Verify template files are in the correct directory
4. **Dependency issues**: Ensure all packages in **requirements.txt** are installed

**Next Development Steps**

1. Implement user authentication
2. Add training content management
3. Create detailed reporting features
4. Implement email notifications
5. Add multi-language support

This document should be saved as **docs/setup\_guide.md** in your project folder for future reference.