

Bing

I apologize for the confusion. Here's a more detailed example of a Flask web application, a build script, and test cases.

Source Code:

```
# app.py
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return "Welcome to China Bank!"

if __name__ == '__main__':
    app.run(debug=True)
```

Build Script:

```
# build.sh
echo "Installing dependencies..."
pip install -r requirements.txt

echo "Starting application..."
python app.py
```

Test Cases:

```
# test_app.py
import unittest
import app

class FlaskTestCase(unittest.TestCase):

    def setUp(self):
        self.app = app.app.test_client()

    def tearDown(self):
        pass

    def test_home(self):
        response = self.app.get('/', follow_redirects=True)
        self.assertEqual(response.status_code, 200)
        self.assertEqual(response.data, b'Welcome to China Bank!')

if __name__ == '__main__':
    unittest.main()
```

In this case, requirements.txt should contain:

```
flask==1.1.2
```

To run the application, you would run the build script (build.sh). This installs the necessary dependencies and starts the application. The test cases can be run with the command `python -m unittest test_app.py`.

Please note that this is a very basic example and doesn't include any fintech-specific functionality. You'll need to add routes, templates, and possibly a database to store and manipulate financial data. You might also need to integrate with other services, like payment gateways or financial data providers, depending on the specific requirements of your client.

Remember to always follow best practices for web development, including proper handling of user input and errors, and protecting sensitive data.