Test case T001

Test number	T001	Test name	Deployment
Description	Create a new deployment of web app and database		
Prerequisite tests		Test creator	Jakub Novák
Tests requirements from assignment	1, 2, 3, 5, 7, 8, 9	Test creation date	2026-01-04

Steps

1. Connect to Microsoft SQL Server using SQL Server Management Studio.
2. Start the SQL Server Agent service if it is not running.
   - See https://database.guide/enable-sql-server-agent-via-ssms/ for a guide.
3. Create a new query.
4. Paste script within `database.sql` into the newly created query.
5. Confirm that the table definitions in the query contain colums with types `decimal`, `bit`, `varchar`, `datetime` and that some columns have a check constraint in the form of `check(<column> in ('<str1>', '<str2>', ...))`.
6. Confirm that the `payment` table contains at least two foreign keys referencing one table.
   - These are written as `foreign key references <table>(id)`.
7. Confirm that there exist at least two view definitions in the form of `create view <view_name> ...`.
8. Confirm that transactions are used.
   - Transactions are denoted with `begin transaction`, `commit transactions` and `rollback`.
9. Execute query.
10. Reload `Databases` in *Object Explorer*.
11. Confirm that a database named `egg_bank` was created.
12. Confirm that the databae contains tables `users`, `account`, `payment`, `log` and `log_msg_type`.
13. Install dependencies by executing `npm install` in a console in the project directory.
14. Update configuration options in the `.env` file in project root directory.
    - See *README.md* for configuration options.
15. Run the server by executing `npm run dev` in a console in the project directory.
16. Change configuration to something incorrect and try to restart server.
    - Server can be stopped by pressing CTRL-C.
17. Confirm that server does not start and prints an error message.
18. Open Web UI in a browser at the address `http://localhost:5173`.
19. Open *Login* page.
20. Log in with name `admin` and password `Hunter12`.
21. Click on *Import data* button.
22. For the first file input select `users.json` from the `export/` directory.
23. For the second file input select `account.json` from the `export/` directory.
24. For the third file input select `payment.json` from the `export/` directory.
25. Click on *Import*.

Confirmation of expected results

1. Click on button labeled *Back to dashboard*.

2. Confirm that log section contains messages of user creation and opening of accounts.
3. Click on button labeled *Logout*.
4. Click on button labeled *Login*.
5. Enter name `karel` and password `karel`.
6. Confirm that user *karel* has two accounts opened one of type *basic* and the other of type *savings*.
   - Under account name there should be written either `basic` or `savings`.
7. Confirm that in both accounts there are some eggs stored.
   - Under account type there should be a number greater than 0 with an egg (🥚) after.