Test plan

Testing range：

Functional requirement:

1. All web pages.
2. Databases

Non-functional requirement:

1. Security test
2. Compatible test
3. Performance test

Testing tools:

1. selgium: automate functional test and unit test
2. Jmeter: performance test
3. Redmine: record defects of implementation discovered by members

Testing management scheme:

The quality information of project should be recorded in wiki every week.

Example

In the process of software testing, bugs are discovered and solved which can improve the quality and make our web application reliable. Software testing uses a variety of testing approaches, like unit testing, integration testing, functional testing, and browser compatibility testing to ensure the application is bug-free and requirement is fully implemented.

Browser compatible test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | IE | Chrome | Edge | Firefox |
| Font | √ | √ | √ | √ |
| Table | √ | √ | √ | √ |
| Navigation bar | √ | √ | √ | √ |
| Form | √ | √ | √ | √ |

Function test:

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Test case | Expects | Results |
| Create a task | 1. In homepage, select menu New a task 2. In create task page, fill task information   Click create button | 1. Create task page is displayed  2. Task event is created and items has been successfully added into table.  3.After user add a task, the page should go back to index.html | pass |
| Show to do list | 1. In homepage, all uncompleted tasks is displayed. 2. In viewTask page, the completed and uncompleted tasks is displayed separately. | 1. All information is correct. | pass |
| Mark task as complete | 1. In homepage, select one task and mark it as complete.(click the complete button) | 1. The task should disappear on this page and in viewTask page, the task appear in complete table. | pass |
| Edit task | 1. In homepage, select one task and click the edit button 2. In edit page, fill task information 3. Click save change button. 4. User changes the title into a new one that has existed. | 1. The page jump to edit page correctly. 2. After the user submit the form, the page should go back to homepage. 3. The information has been changed and displayed correctly in homepage. 4. A flash message will warn user of repeated title. | pass |
| Error page | 1. In homepage, select menu New a task 2. Fill the form with existed title or title longer than 16 words 3. Click create button 4. Modify URL in browser, which page is not existed | 1. The homepage gives out a flash message to warn user of invalid input. 2. The “page not found” error page should be displayed | pass |
| Script to manage database migrate, upgrade and rollback | 1. Add index to id in data model 2. Execute db upgrade   Execute db downgrade | 1. After run db upgrade, We can find index added in db structure   After run db downgrade, we can find index is disappered | pass |

Database migration, updates and rollbacks test case running detail is as follow:

1. Create migrate warehouse

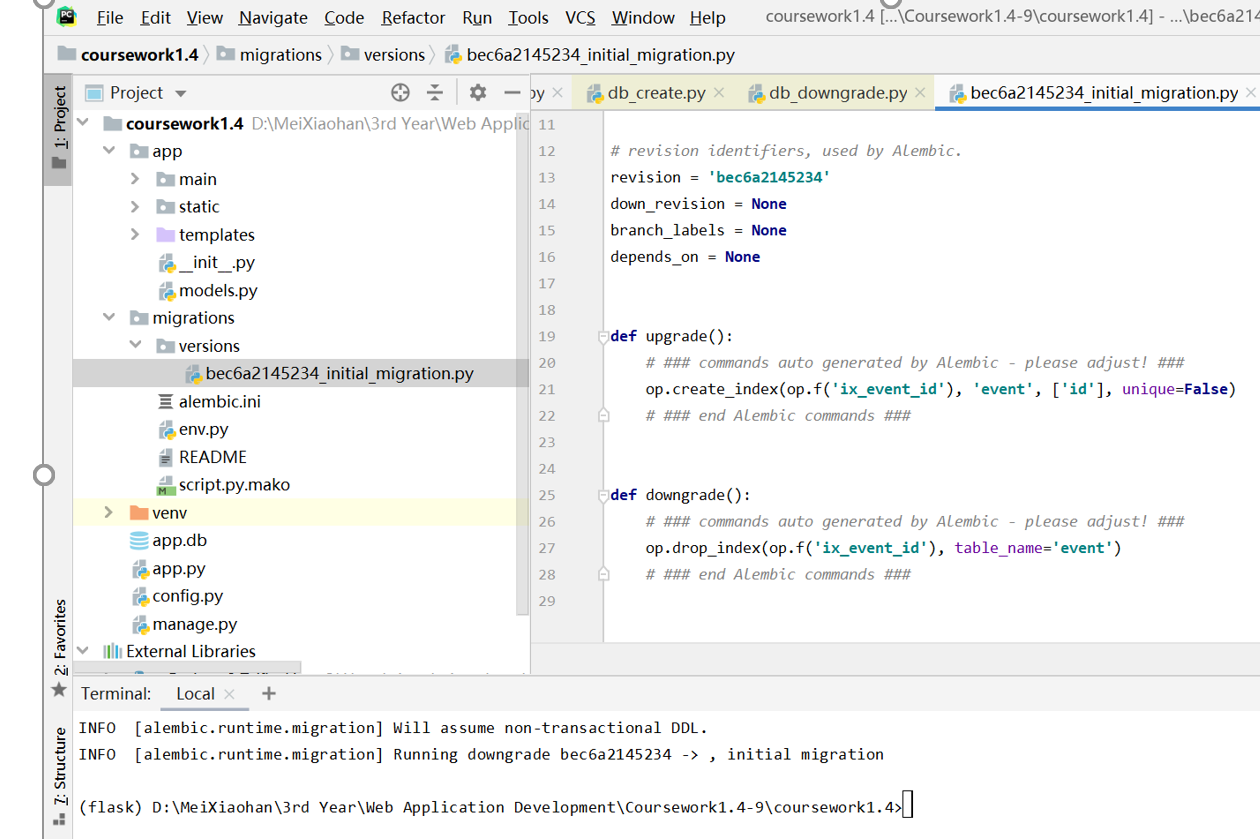
>python manage.py db init

1. Add index in id

id = db.Column(db.Integer, primary\_key=**True**, index=**True**)

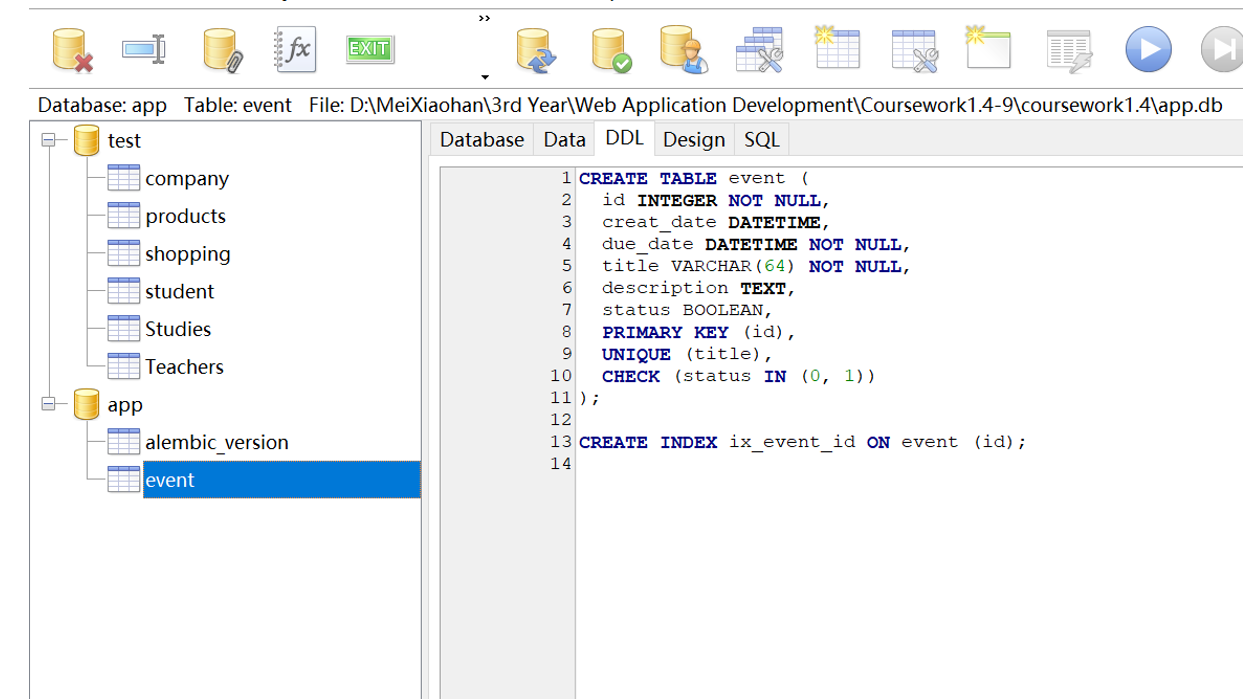
1. Automatically Create migrate script：

>python manage.py db migrate - m "initail migration"



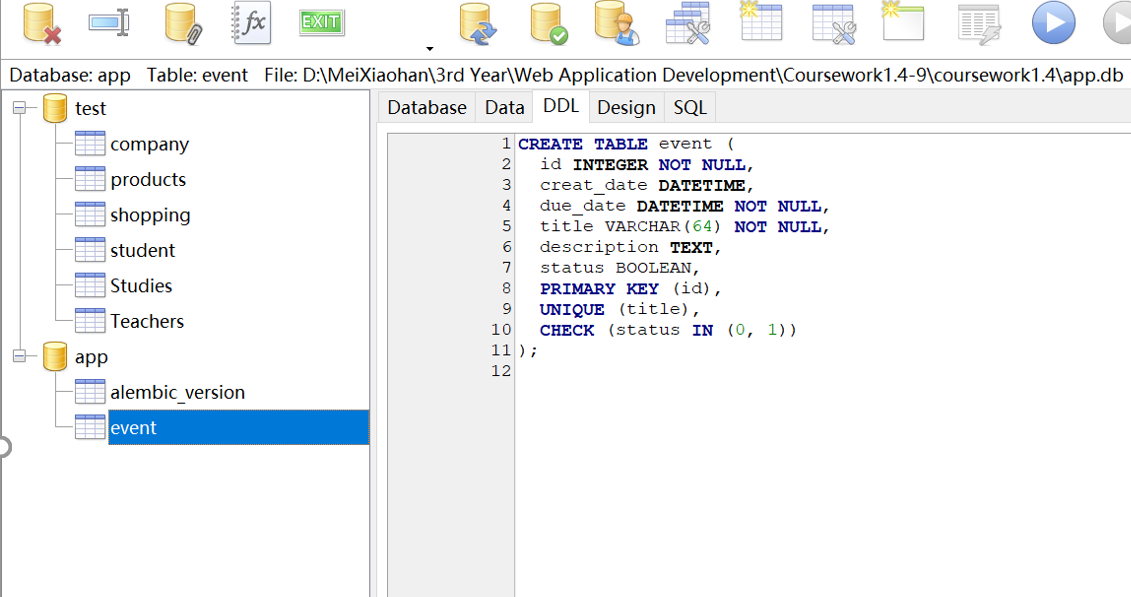
1. Update DB：

>python manage.py db upgrade



1. Rollback DB

>python manage.py db downgrade



The compatibility test and function test proves that this web application can display well on most of the browsers and its function can satisfy the specification.