**Nette Notes App - Step-by-Step Development Guide**

### 1. Create a New Nette Project

composer create-project nette/web-project nette-notes-app  
cd nette-notes-app

### 2. Configure Database Connection

In app/config/local.neon, set the DB connection:

database:  
 dsn: 'mysql:host=127.0.0.1;dbname=nette\_notes'  
 user: your\_db\_user  
 password: your\_db\_password

### 3. Prepare the Database Schema

Create a SQL file sql/init.sql with:

CREATE TABLE users (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 email VARCHAR(255) NOT NULL UNIQUE,  
 password VARCHAR(255) NOT NULL  
);  
  
INSERT INTO users (email, password)  
VALUES ('admin@demo.cz', '$2y$10$examplehashedpassword'); -- hash of heslo123  
  
CREATE TABLE notes (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 user\_id INT NOT NULL,  
 title VARCHAR(100) NOT NULL,  
 content TEXT NOT NULL,  
 created\_at DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE  
);

Import to MySQL:

mysql -u your\_user -p nette\_notes < sql/init.sql

### 4. Implement the Authenticator

Create app/Model/UserAuthenticator.php:

namespace App\Model;  
  
use Nette\Security\Authenticator;  
use Nette\Security\SimpleIdentity;  
use Nette\Security\AuthenticationException;  
use Nette\Database\Explorer;  
  
class UserAuthenticator implements Authenticator  
{  
 private Explorer $db;  
  
 public function \_\_construct(Explorer $db)  
 {  
 $this->db = $db;  
 }  
  
 public function authenticate(string $user, string $password): SimpleIdentity  
 {  
 $row = $this->db->table('users')->where('email', $user)->fetch();  
 if (!$row || !password\_verify($password, $row->password)) {  
 throw new AuthenticationException('Invalid credentials.');  
 }  
 return new SimpleIdentity($row->id, [], ['email' => $row->email]);  
 }  
}

Register it in app/config/common.neon:

services:  
 - App\Model\UserAuthenticator

### 5. Create LoginPresenter

class LoginPresenter extends Nette\Application\UI\Presenter  
{  
 protected function createComponentLoginForm(): Form  
 {  
 $form = new Form;  
 $form->addEmail('email', 'Email:')->setRequired();  
 $form->addPassword('password', 'Password:')->setRequired();  
 $form->addSubmit('send', 'Login');  
 $form->onSuccess[] = [$this, 'loginFormSucceeded'];  
 return $form;  
 }  
  
 public function loginFormSucceeded(Form $form, \stdClass $values): void  
 {  
 try {  
 $this->getUser()->login($values->email, $values->password);  
 $this->redirect('Note:default');  
 } catch (AuthenticationException $e) {  
 $form->addError('Invalid login.');  
 }  
 }  
  
 public function actionLogout(): void  
 {  
 $this->getUser()->logout();  
 $this->redirect('Login:default');  
 }  
}

Template: templates/Login/default.latte

<h2>Login</h2>  
{form loginForm}  
 <div>{input email}</div>  
 <div>{input password}</div>  
 <div>{input send}</div>  
{/form}

### 6. Create NotePresenter

class NotePresenter extends Nette\Application\UI\Presenter  
{  
 public function startup(): void  
 {  
 parent::startup();  
 if (!$this->getUser()->isLoggedIn()) {  
 $this->redirect('Login:default');  
 }  
 }  
  
 public function renderDefault(): void  
 {  
 $this->template->notes = $this->db->table('notes')->where('user\_id', $this->getUser()->getId());  
 }  
  
 public function actionDelete(int $id): void  
 {  
 $note = $this->db->table('notes')->get($id);  
 if ($note && $note->user\_id === $this->getUser()->getId()) {  
 $note->delete();  
 }  
 $this->redirect('default');  
 }  
  
 protected function createComponentNoteForm(): Form  
 {  
 $form = new Form;  
 $form->addText('title', 'Title:')->setRequired()->addRule(Form::MAX\_LENGTH, 'Max 100 chars', 100);  
 $form->addTextArea('content', 'Content:')->setRequired()->addRule(Form::MIN\_LENGTH, 'Min 10 chars', 10);  
 $form->addSubmit('send', 'Save');  
 $form->onSuccess[] = [$this, 'noteFormSucceeded'];  
 return $form;  
 }  
  
 public function noteFormSucceeded(Form $form, \stdClass $values): void  
 {  
 $this->db->table('notes')->insert([  
 'user\_id' => $this->getUser()->getId(),  
 'title' => $values->title,  
 'content' => $values->content,  
 ]);  
 $this->redirect('default');  
 }  
}

Template: templates/Note/default.latte

<h2>Your Notes</h2>  
<ul>  
{foreach $notes as $note}  
 <li><strong>{$note->title|escapeHtml}</strong> — {$note->content|escapeHtml}  
 <a n:href="delete, id => $note->id">Delete</a>  
 </li>  
{/foreach}  
</ul>  
  
<h3>Add Note</h3>  
{form noteForm}  
 <div>{input title}</div>  
 <div>{input content}</div>  
 <div>{input send}</div>  
{/form}

### 7. Routes

In app/Router/RouterFactory.php:

$router->addRoute('login', 'Login:default');  
$router->addRoute('logout', 'Login:logout');  
$router->addRoute('<presenter>/<action>[/<id>]', 'Note:default');

### 8. Run the Application

php -S localhost:8000 -t www

Go to: <http://localhost:8000>

Login with:

* **Email:** admin@demo.cz
* **Password:** heslo123

### Done!

You now have a functional Nette application with login and note management.