

QA/Test Strategy + Test Evidence

۴/۵/۱ مقدمه

به عنوان یک سرویس حساس پزشکی `nobatnou.ir` سیستم نوبتدهی نیازمند سطح بالایی از قابلیت اطمینان، در دسترس بودن و امنیت است. تضمین و اجرای منظم انواع (QA Strategy) کیفیت از طریق استراتژی جامع تست تست‌ها انجام می‌شود.

۴/۵/۲ Aهداف QA**

کارکرد صحیح: ** همه سناریوهای کاربری طبق نیازمندی‌ها اجرا شوند** -

قابلیت اطمینان: ** سیستم در شرایط واقعی پایدار باشد** -

امنیت: ** داده‌های کاربران و فرآیندهای حساس محافظت شوند** -

تجربه کاربری: ** رابط کاربری روان، بدون باگ و قابل درک باشد** -

قابلیت نگهداری: ** کد تمیز، قابل تست و مستند باشد** -

۴/۵/۳ انواع تست‌های اجرایی

۱. تست واحد (Unit Testing)

هدف: **بررسی عملکرد هر تابع یا متدهای صورت مستقل** -

(برای جاوااسکریپت) PHPUnit + Jest (برای PHP) چارچوب** -

و ۷۰٪ جاوااسکریپت PHP پوشش هدف: **حداقل ۸۰٪ کد** -

موارد تست شده -

- `add_nobat_ajax` ، `login_user_ajax` ،
`register_user_ajax`

اعتبارسنجی شماره تلفن (۱۱ رقم، فقط عدد) -

(`position` محاسبه موقعیت نوبت -

هش رمز عبور و احراز هویت -

- **ابزار اندازهگیری**: Xdebug + PHPUnit coverage report.

۲. تست یکپارچه‌سازی (Integration Testing)

هدف: **تست تعامل بین کامپوننت‌ها** - (AJAX، سشن، دیتابیس)

موارد تست -

- ثبت نوبت → ذخیره در دیتابیس → نمایش در لیست نوبت‌ها -
ورود کاربر → ایجاد سشن → دسترسی به داشبورد -
لغو نوبت → بهروزرسانی وضعیت در دیتابیس -
- روش: ** استفاده از دیتابیس تست** و MySQL mock session.

۳. end-to-end (E2E)

- هدف: ** شبیه‌سازی رفتار واقعی کاربر از آغاز تا پایان** -
- چارچوب: ** Cypress یا Playwright. -
- سناریوهای کلیدی: **

1. کاربر جدید ثبت‌نام می‌کند → وارد می‌شود → نوبت جدید می‌گیرد.
2. کاربر نوبت خود را پیگیری می‌کند (وضعیت و زمان تخمینی).
3. ادمین وضعیت نوبت را تغییر می‌دهد (pending → in-progress → completed).
4. خود را لغو می‌کند کاربر نوبت با داده‌های ساختگی staging محیط: ** محیط -

۴. تست امنیت (Security Testing)

- هدف: ** شناسایی آسیب‌پذیری‌های امنیتی *

- **:موارد بررسی*

- استفاده شده‌اند `query` ها با SQL همه: \$wpdb->prepare تزریق -
- XSS پالایش `sanitize_text_field` و `esc_html` خروجی‌ها با می‌شوند.
- CSRF (بررسی سشن) نیاز به افزودن) بررسی توکن غیرفعال: wp_nonce).
- احراز Ajax (بررسی سشن) محدود شده (دسترسی به دسترسی به - ها حساس: رمز عبور با - هش می‌شود `wp_hash_password` داده‌های `wp_hash_password` ابزار** + (اسکن خودکار) OWASP ZAP دستی کد -

۵. (Regression Testing)**

- هدف: **اطمینان از عدم شکست قابلیت‌های موجود پس از تغییرات** - روش** -

اجرای خودکار تست‌های واحد و یکپارچه در هر commit (GitHub Actions).

قبل از هر انتشار E2E اجرای دستی سناریوهای -

های اصلی (ثبت‌نام، ورود، نوبت‌گیری، مدیریت feature پوشش: همه** - نوبت)

۶. (Load Testing)**

. هدف: **بررسی عملکرد تحت بار کاربری زیاد** -

. سناریو: **۱۰۰ کاربر همزمان درخواست نوبت ارسال می‌کنند** -

- **ابزار Apache JMeter یا k6.

- **معیارها**

. زمان پاسخگویی زیر ۲ ثانیه -

. نرخ خطا زیر ۱٪ -

. حداقل ۵۰ توان پردازش همزمان -

* و تضمین کیفیت release فرآیند ۴/۵/۴ ***

1. جدا ایجاد branch جدید را در feature توسعه: ** توسعه دهنده** می‌کند.

2. GitHub Actions. تست واحد/بیکارچه: ** اجرای خودکار با**.

3. **(Code Review) مرور کد را دیگر کد را توسعه دهنده یک بررسی می‌کند.

4. **staging:** QA تست دستی در کلیدی را اجرا می‌کند.

5. ** اجرای کامل** تست رگرسیون: ** تست موجود Suite.

6. ** اسکن خودکار + بررسی دستی آسیب‌پذیری‌ها**. تست امنیت:

تصویب انتشار: ** مدیر فنی تأیید نهایی می‌کند** 7.

8. ** production:** deployment با rollback آماده.

9. ** Sentry, Google Analytics. رصد خطاها و عملکرد با ابزارهایی مانند** مانیتورینگ

۴/۵/۵ مستندات تست (Test Evidence)**

گزارش اجرای تست (نمونه) ***

تست نوع	تاریخ اجرا	تعداد تست‌ها	گذرنده	شکست خورده	پوشش کد
---------	------------	--------------	--------	------------	---------

----- ----- ----- ----- ----- -----
----- ----- ----- ----- ----- -----

(PHP) واحد	۱۵/۰۲/۱۴۰۳	۴۲	۴۰	۲
			۷۸%	

(JS) واحد	۱۵/۰۲/۱۴۰۳	۱۸	۱۸	۰
			۶۵%	

۱	۱۱	۱۲	۱۶/۰۲/۱۴۰۳	بکارچهسازی	-

E2E	١٧/٠٢/١٤٠٣	٨	٨	.	-	
١	٤	٥	١٨/٠٢/١٤٠٣	امنیت	-	

نمونه تست واحد **PHPUnit)**

```php

```
class NobatAjaxTest extends WP_UnitTestCase {
 public function
 test_add_nobat_with_valid_phone() {
 $_POST['patient_name'] = 'علی محمدی';
 $_POST['phone_number'] = '09121234567';

 ob_start();
 add_nobat_ajax();
 $output = ob_get_clean();

 $this->assertStringContainsString('نوبت ثبت شد',
 $output);
 }
}
```

}

public function  
test\_add\_nobat\_with\_invalid\_phone() {

\$\_POST['patient\_name'] = 'على محمدى';  
\$\_POST['phone\_number'] = '123'; // اشتباه

ob\_start();

add\_nobat\_ajax();

\$output = ob\_get\_clean();

شماره تلفن باید ۱۱ رقم (' ۱۱ رقم' 'باشد', \$output);

}

}

...

##### \*\* گزارش پوشش کد (Coverage Report)\*\*

[تصویر گزارش پوشش کد] (<https://nobatnou.ir/coverage-report-14030215.png>)

\* (لینک نمونه به گزارش واقعی)

---

## # ## \*\*۴/۵/۶\*\* سیاست رفع باگ (Bug Fix Policy)\*\*

1. \*\*طبقه‌بندی باگ‌ها\*\*

- \*\*Critical:\*\* سیستم از کار افتاده (مثلاً ثبت نوبت امکان‌پذیر نیست).

زمان رفع: حداقل ۲ ساعت

- \*\*High:\*\* عملکرد اصلی مختل شده (مثلاً محاسبه زمان تخمینی اشتباه).

زمان رفع: ۲۴ ساعت

- \*\*Medium:\*\* مشکل در بخش غیرحساس (مثلاً نمایش نادرست در مرورگر خاص).

زمان رفع: ۷ روز

- \*\*Low:\*\* مشکل ظاهری جزئی

زمان رفع: در انتشار بعدی

2. \*\*فرآیند رفع باگ\*\*

ثبت می‌شود GitHub Issues گزارش باگ در -

برچسب‌گذاری و اختصاص به توسعه‌دهنده -

- برای رفع branch(`fix/bug-description`).

نوشتن تست برای باگ (حتی المقدور) -

- رفع کد → اجرای تست‌ها → مرور کد -

در صورت فوریت patch انتشار -

### 3. \*\*پیگیری:\*\*

قابل رصد GitHub Projects (مثلاً board پروژه) همه باگ‌ها در هستند.

---

### \*\*نتیجه‌گیری ۷/۵/۴\*\*

بر پایه \*\*تست خودکار\*\*، \*\*مرور`nobatnou.ir` در QA استراتژی جدید تنها پس از feature کد\*\* و \*\*مانیتورینگ مستمر\*\* استوار است. هر گذراندن مراحل فوق اجازه انتشار می‌یابد. این فرآیند از شکست سیستم در تولید جلوگیری کرده و هزینه نگهداری را کاهش می‌دهد.