

# رودمپ مدیریت پروژه تیم توسعه‌دهنده وب

## مرحله آغازین (Project Initiation)

در این فاز هدف پروژه مشخص می‌شود. نیازمندی‌های کارفرما جمع‌آوری شده، محدوده و خروجی‌های مورد انتظار تعیین می‌گردد. همچنین تکنولوژی‌های اصلی (زبان‌ها و فریمورک‌ها) برای فرانت‌اند و بکاند انتخاب می‌شوند و نقش‌های تیمی شامل توسعه‌دهنگان فرانت‌اند، بکاند، طراحی، UI/UX و QA تعریف خواهد شد.

## مرحله برنامه‌ریزی (Planning)

در این بخش ویژگی‌ها و وظایف بین تیم فرانت و بک تقسیم می‌شود. معماری کلی پروژه (ساختار دیتابیس، API‌ها و لایه‌های نرم‌افزار) طراحی شده و Sprint‌ها و Milestone‌ها برای مدیریت زمان‌بندی تعریف می‌گردند. همچنین طراحی اولیه رابط کاربری (Wireframe و Prototype) آماده می‌شود تا همه تیم روی یک مسیر مشخص حرکت کنند.

## مرحله توسعه (Development)

فرانت‌اند: توسعه رابط کاربری، طراحی صفحات و کامپوننت‌ها، رعایت اصول Responsive و اتصال به API‌های بک اند.

بکاند: پیاده‌سازی منطق تجاری (Business Logic)، ایجاد API‌ها، طراحی دیتابیس، و نوشتن Unit Test برای بخش های حیاتی سیستم.

## کنترل نسخه و همکاری تیمی (Version Control & Collaboration)

کلیه کدنویسی‌های روی GitHub قرار می‌گیرد. برای مدیریت بهتر کد، Branch Strategy (main, dev, feature) می‌شود و تمام Pull Request‌ها قبل از ادغام، Code Review خواهند شد. همچنین CI/CD branches روی مخزن تنظیم می‌گردد تا تست‌ها و Workflow Build‌ها به‌طور خودکار انجام شوند.

## مرحله تست و تضمین کیفیت (Testing & QA)

در این مرحله تیم QA و توسعه‌دهنگان تست‌های مختلف انجام می‌دهند: Unit Test برای مازول‌ها Integration Test برای ارتباط بین اجزای سیستم User Acceptance Test برای بررسی محصول از دید کاربر گزارش باگ‌ها ثبت و اصلاح می‌شوند تا کیفیت نهایی تضمین شود.

## مرحله استقرار (Deployment)

کدهای نهایی از طریق CI/CD از GitHub به VPS منتقل می‌شوند. روی سرور، سرویس‌ها با ابزارهایی مثل Docker یا PM2 مدیریت می‌شوند. برای SSL/TLS و Reverse Proxy Nginx/Apache برای امنیت راهاندازی خواهد شد. همچنین مانیتورینگ اولیه سرور جهت بررسی عملکرد و پایداری صورت می‌گیرد.

## مرحله نگهداری و پشتیبانی (Maintenance & Support)

پس از انتشار، سیستم به طور مداوم مانیتور می‌شود. پشتیبان‌گیری منظم از دیتابیس و کد انجام شده، مشکلات احتمالی رفع می‌گردد و نسخه‌های جدید پروژه (Feature Update) ها منتشر می‌شوند. این فاز تضمین می‌کند که پروژه پایدار و قابل توسعه باقی بماند.

## لایه مدیریت پروژه (Project Management Layer)

برای هماهنگی تیم و کارفرما:

- جلسات روزانه کوتاه (Daily Stand-up) جهت هماهنگی داخلی تیم

- جلسات هفتگی با کارفرما برای ارائه گزارش پیشرفت

- استفاده از ابزار مدیریت پروژه (مانند Jira یا Trello) جهت پیگیری وظایف و تحويلها

- مستندسازی کامل پروژه برای استفاده تیم و کارفرما

## سؤالات و پاسخ‌ها

### ۱- اگر سایت پویا باشد مشکلات آن از کجا معلوم می‌شود؟

سایت پویا چون وابسته به دیتابیس و کدنویسی بکاند هست، مشکلاتش معمولاً در سه بخش مشخص می‌شود:

- لاغ‌ها (Logs) که خطاهای سرور و برنامه را ذخیره می‌کنند.

- تست‌های خودکار (Unit Test, Integration Test) و تست کاربری.

- مانیتورینگ و ابزارهای گزارش خطای مثل Sentry یا Grafana

همچنین بازخورد کاربران هم می‌تواند مشکلات را نشان دهد.

### ۲- مدیر پروژه چطور می‌تواند کدها و اجرای آنها را قبل از اینکه منتشر بشوند مشاهده کند؟

مدیر پروژه می‌تواند از روش‌های زیر استفاده کند:

- بررسی کدها در GitHub Pull Request های

- مشاهده نسخه آزمایشی در محیط Staging/Testing Server قبل از انتشار.

- دریافت دموهای داخلی از تیم توسعه و تست عملی محصول.

این روش‌ها کمک می‌کند قبل از انتشار، محصول بررسی و تأیید شود.

### ۳- وظیفه CI/CD چیه؟

Continuous Integration / Continuous Deployment (CI/CD) یعنی

- CI : هر بار کدی اضافه شود، به طور خودکار Build و تست اجرا می‌شود.

- CD : در صورت موفقیت تست‌ها، کد به طور خودکار روی سرور منتشر می‌شود.

مزایا : شناسایی سریع خطاهای افزایش سرعت و دقت انتشار، و اطمینان از سالم بودن نسخه نهایی روی سرور.