

# رودمپ مدیریت پروژه تیم توسعه 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 توسعه‌دهندگان فرانت‌اند، بک‌اند، طراح UI/UX، DevOps و QA تعریف خواهد شد.

## مرحله برنامه‌ریزی (Planning)

در این بخش ویژگی‌ها و وظایف بین تیم فرانت‌اند و بک‌اند تقسیم می‌شود. معماری کلی پروژه (ساختار دیتابیس، API ها و لایه‌های نرم‌افزار) طراحی شده و Milestone ها و Sprint ها 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 API های بک‌اند.

بک‌اند: پیاده‌سازی منطق تجاری (Business Logic)، ایجاد API ها، طراحی دیتابیس، و نوشتن Unit Test برای بخش‌های حیاتی سیستم.

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

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

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

در این مرحله تیم QA و توسعه‌دهندگان تست‌های مختلف انجام می‌دهند:

- Unit Test- برای ماژول‌ها
- 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. Nginx/Apache برای Reverse Proxy و SSL/TLS 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 به‌طور مداوم مانیتور می‌شود. پشتیبان‌گیری منظم از دیتابیس و کد انجام شده، مشکلات احتمالی رفع می‌گردند 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 تحویل‌ها
- مستندسازی کامل پروژه برای استفاده تیم و کارفرما

## سوالات و پاسخ‌ها

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

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

- لاگ‌ها (Logs) که خطاهای سرور و برنامه را ذخیره می‌کنند.
- تست‌های خودکار (Unit Test, Integration Test) و تست کاربری.
- مانیتورینگ و ابزارهای گزارش خطا مثل Sentry یا Grafana

همچنین بازخورد کاربران هم می‌تواند مشکلات را نشان دهد.

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

مدیر پروژه می‌تواند از روش‌های زیر استفاده کند:

- بررسی کدها در Pull Request های GitHub.
- 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 چیست؟

CI/CD یعنی Continuous Integration / Continuous Deployment

- CI : هر بار کدی اضافه شود، به‌طور خودکار Build و تست اجرا می‌شود.
  - CD : در صورت موفقیت تست‌ها، کد به‌طور خودکار روی سرور منتشر می‌شود.
- مزایا : شناسایی سریع خطاها، افزایش سرعت و دقت انتشار، و اطمینان از سالم بودن نسخه نهایی روی سرور.