

Work Breakdown Structure (WBS) - Insurance Website (ASP.NET MVC)

1. Project Management

- 1.1 Project planning and kickoff
- 1.2 Team & role assignments
- 1.3 Sprint planning & milestone definition
- 1.4 Risk and issue tracking
- 1.5 Weekly meetings and progress tracking
- 1.6 Final review and closure

2. Requirements & Design

- 2.1 Business requirements gathering
 - 2.1.1 Stakeholder interviews
 - 2.1.2 Feature clarification with team
- 2.2 System design
 - 2.2.1 MVC architecture planning
 - 2.2.2 Database schema design (Users, Policies, Claims, Payments, etc.)
 - 2.2.3 Role-based access planning
- 2.3 UI/UX design
 - 2.3.1 Wireframes for main pages
 - 2.3.2 High-fidelity mockups
 - 2.3.3 Client/team feedback

3. Frontend Development (Views - Razor + Bootstrap)

- 3.1 Public pages
 - 3.1.1 Home page
 - 3.1.2 About / Contact / FAQ
 - 3.1.3 Insurance product details (Car, Health, Life)
- 3.2 User portal
 - 3.2.1 User registration & login (ASP.NET Identity)
 - 3.2.2 View & edit profile
 - 3.2.3 Request insurance
 - 3.2.4 View active/past policies
 - 3.2.5 Submit and track claims
 - 3.2.6 View wallet & payment history
- 3.3 Company portal
 - 3.3.1 View incoming insurance requests
 - 3.3.2 Respond to requests (approve/decline)
 - 3.3.3 Manage policies
 - 3.3.4 Process claims
- 3.4 Admin panel
 - 3.4.1 Manage users
 - 3.4.2 Manage companies
 - 3.4.3 View all insurance activity (requests, claims, policies)
 - 3.4.4 Control system settings, roles, and permissions

4. Backend Development (Controllers & Models - MVC)

- 4.1 Identity & Role management
 - 4.1.1 User, Company, Admin roles
 - 4.1.2 Authorization checks in controllers
- 4.2 Insurance requests module
 - 4.2.1 Submit request
 - 4.2.2 View/approve/decline requests (company/admin)

4.3 Policies management

■ ■ 4.3.1 Create/assign policy

■ ■ 4.3.2 Policy expiry, renewal, status logic

4.4 Claims processing

■ ■ 4.4.1 Claim submission

■ ■ 4.4.2 Review and approve/reject (by company)

■ ■ 4.4.3 Compensation logic

4.5 Payments module (Stripe)

■ ■ 4.5.1 User wallet top-up

■ ■ 4.5.2 Auto-deduct subscription fees

■ ■ 4.5.3 Monthly payout to company minus 15% commission

4.6 Notifications (Email / In-app if needed)

5. Admin Dashboard (Web)

5.1 Dashboard overview (statistics, graphs)

5.2 Manage users and companies

5.3 View all policies and requests

5.4 Manual intervention (force approval, override decisions)

5.5 Financial reports (commission, Stripe income)

5.6 Access control and logs

6. Testing & QA

6.1 Unit testing (models and logic)

6.2 Functional testing (form submissions, business flow)

6.3 Role-based testing (user, company, admin)

6.4 Payment flow testing (Stripe test keys)

6.5 UAT (User Acceptance Testing)

6.6 Bug tracking and resolution

7. Deployment & Launch

7.1 Configure server and deploy backend

7.2 Configure domain and SSL

7.3 Stripe live mode configuration

7.4 Final production testing

7.5 Launch and monitoring

8. Documentation & Training

8.1 User guides (how to request insurance, file claims, etc.)

8.2 Admin & company documentation

8.3 Developer documentation (folder structure, important flows)

8.4 Training session for support/admin staff