

Metode și tehnici de dezvoltare a aplicației

Versiune document: 2 iulie 2026

Metode și tehnici de dezvoltare a aplicației FiscontExpert Consult

Versiune document: 2 iulie 2026 Tip document: specificație tehnică Limbaj: tehnic Aplicație: Next.js full-stack, Prisma, MySQL/MariaDB, cPanel Node.js

1. Metodologii SDLC aplicabile

1.1. Scrum

Artefacte:

- Product Backlog;
- Sprint Backlog;
- Increment;
- Definition of Ready;
- Definition of Done;
- Burndown/Burnup;
- Sprint Review report;
- Sprint Retrospective report.

Ceremonii:

- Sprint Planning;
- Daily Scrum;
- Backlog Refinement;
- Sprint Review;
- Sprint Retrospective.

Aplicare:

- dezvoltare iterativ-incrementală;
- prioritizare pe epice/module;
- livrare incrementală pe sprinturi;
- validare UI/UX și funcțională la final de sprint;
- gestiune schimbări prin backlog.

1.2. Waterfall

Faze:

- inițiere;

- colectare cerințe;
- analiză funcțională;
- analiză tehnică;
- design UI/UX;
- arhitectură;
- implementare;
- testare;
- stabilizare;
- acceptanță;
- deploy;
- mentenanță.

Aplicare:

- documentație formală;
- audit;
- GDPR/DPIA/GAP;
- contractare;
- acceptanță finală;
- livrare pe faze cu gate-uri.

1.3. Kanban

Coloane:

- Backlog;
- Ready;
- In Progress;
- Blocked;
- Review;
- Testing;
- Done.

Aplicare:

- mentenanță;
- bug fixing;
- operațiuni live;
- suport hosting;
- corecturi punctuale.

1.4. Hybrid Agile-Waterfall

Structură:

- Waterfall pentru cerințe, audit, documentație, livrare formală;

- Scrum pentru implementare;
- Kanban pentru bug fixing și mentenanță.

Aplicare recomandată:

Waterfall → definire și acceptanță
 Scrum → dezvoltare iterativă
 Kanban → mentenanță și suport live

2. Analiză cerințe

2.1. Requirement Engineering

Tehnici:

- elicitation;
- stakeholder mapping;
- user story mapping;
- use case modeling;
- domain modeling;
- acceptance criteria definition;
- non-functional requirements analysis;
- traceability matrix.

Artefacte:

- user stories;
- epics;
- use cases;
- business rules;
- technical constraints;
- acceptance criteria;
- risk register;
- compliance checklist.

2.2. Functional Requirements

Categorii:

- site public;
- CMS;
- template engine;
- i18n;
- authentication;
- authorization;
- portal client;
- document management;

- financial analysis;
- economic analysis;
- SWOT;
- GDPR client analysis;
- admin dashboard;
- payments;
- email marketing;
- chat;
- external integrations;
- GDPR/DPIA/GAP reporting;
- retention/erasure;
- backup/restore;
- maintenance mode.

2.3. Non-Functional Requirements

Categorii:

- security;
- performance;
- availability;
- scalability;
- maintainability;
- observability;
- accessibility;
- SEO;
- privacy;
- compliance;
- portability;
- recoverability;
- compatibility;
- responsive layout.

3. Arhitectură aplicație

3.1. Stil arhitectural

Tip:

Full-stack monolithic web application

Stack:

Next.js App Router
 React Server Components
 Server Actions
 API Routes

Prisma ORM
MySQL/MariaDB
Node.js custom server
cPanel Passenger runtime

3.2. Separare logică MVC-like

Strat	Implementare
Model	Prisma schema, Prisma Client, <code>src/lib/*</code>
Controller	Server Actions, API routes
View	React components, Next pages/layouts
Services	<code>src/lib/*</code> domain services
Infrastructure	<code>server.js</code> , storage, deployment scripts

3.3. Module boundaries

Module:

- public site module;
- authentication module;
- admin module;
- portal module;
- CMS module;
- template module;
- SEO module;
- compliance module;
- financial analysis module;
- document module;
- payment module;
- email marketing module;
- chat module;
- integration module;
- GDPR erasure module;
- backup module;
- maintenance module.

3.4. Layering

Presentation Layer
↓
Application Layer
↓
Domain Logic Layer
↓
Data Access Layer
↓
Persistence Layer
↓
Infrastructure Layer

4. Design patterns

4.1. Repository-like Data Access

Implementare:

- Prisma Client;
- query-uri centralizate în server actions și servicii;
- acces DB prin model typed.

Aplicare:

- User;
- Company;
- Document;
- SitePage;
- SiteSetting;
- PaymentOrder;
- FinancialAnalysis.

4.2. Service Layer

Implementare:

- `src/lib/financial-analysis.ts`;
- `src/lib/compliance-report.ts`;
- `src/lib/payments.ts`;
- `src/lib/privacy-retention.ts`;
- `src/lib/site-settings.ts`;
- `src/lib/site-templates.ts`.

Rol:

- business logic;
- transformări;
- validări;
- scoruri;
- decizii de retenție;
- generare rapoarte.

4.3. Strategy Pattern

Aplicare:

- template active strategy;
- payment processor strategy;
- email provider strategy;
- integration provider strategy;
- language strategy.

Exemple:

```
TemplateStrategy: original | corporate | netlify  
PaymentProcessorStrategy: netopia | euplatesc | payu  
EmailProviderStrategy: local SMTP | Mailchimp  
IntegrationStrategy: Oblio | SmartBill | TDec | WinMentor | Nexus | Saga
```

4.4. Factory Pattern

Aplicare:

- creare setări default;
- generaere cod easter egg;
- generaere payment order;
- generaere report object;
- generaere financial analysis object.

4.5. Guard Pattern

Aplicare:

- requireUser;
- requireAdmin;
- membership guard;
- document access guard;
- role guard;
- active user guard.

4.6. DTO / FormData Mapping

Aplicare:

- server actions;
- validare input;
- conversie FormData → typed payload;
- sanitizare valori;
- conversie numerică/date/boolean.

4.7. Configuration Object Pattern

Aplicare:

- SiteSetting;
- SEO settings;
- payment settings;
- email settings;
- chat settings;
- security settings;
- integration settings.

4.8. Audit Trail Pattern

Aplicare:

- AuditLog;
- acțiuni admin;
- operații sensibile;
- trasabilitate.

5. Tehnici frontend

5.1. Component-Based UI

Tehnică:

- componente reutilizabile;
- layout public;
- layout admin;
- layout portal;
- widgeturi;
- carduri;
- formulare;
- tabele.

Avantaje:

- reutilizare;
- izolare UI;
- mentenanță;
- testabilitate.

5.2. Responsive Design

Tehnici:

- CSS Grid;
- Flexbox;
- media queries;
- fluid containers;
- relative units;
- image object-fit;
- mobile-first breakpoints;
- overflow control;
- independent scrolling.

Viewporturi țintă:

360x640
390x844
414x896
768x1024

1024x1366
1366x768
1920x1080

5.3. Progressive Enhancement

Aplicare:

- HTML valid;
- formulare server-side;
- fallback fără JS pentru acțiuni critice unde este posibil;
- server-rendered metadata.

5.4. Accessibility Techniques

Tehnici:

- semantic HTML;
- labels pentru inputuri;
- contrast;
- focus state;
- alt text;
- aria-label unde este necesar;
- keyboard navigation;
- responsive readable typography.

5.5. UI State Management

Tipuri:

- server state prin DB;
- form state;
- URL state;
- session state;
- client widget state.

Aplicare:

- chat widget;
- cookie consent;
- template selection;
- form feedback.

6. Tehnici backend

6.1. Server Actions

Utilizare:

- mutații DB;
- formulare admin;

- formulare portal;
- login/logout;
- reset password;
- update settings;
- create analysis;
- GDPR actions.

Avantaje:

- co-locare cu Next.js;
- input server-side;
- control acces;
- redirect/revalidate.

6.2. API Routes

Utilizare:

- endpointuri externe/interne;
- chat;
- ANAF;
- document serving;
- easter egg.

Tehnici:

- method guard;
- JSON validation;
- auth guard;
- access control;
- no-store;
- path traversal protection.

6.3. Input Validation

Tehnici:

- type coercion;
- required fields;
- min/max length;
- enum validation;
- numeric validation;
- date validation;
- email validation;
- MIME validation;
- file size validation.

6.4. Error Handling

Tehnici:

- try/catch;
- controlled error response;
- redirect with status;
- runtime logging;
- no secret leakage;
- generic auth errors;
- audit logging.

6.5. Transactional Consistency

Aplicare recomandată:

- create user + membership;
- payment order + invoice metadata;
- email campaign + recipients;
- erasure request + session invalidation;
- backup snapshot + audit log.

Tehnică:

Prisma transaction

7. Modelare date

7.1. Entity Modeling

Entități:

- User;
- Session;
- Company;
- Membership;
- Document;
- AccountingPeriod;
- FinancialAnalysis;
- SitePage;
- SiteSetting;
- Lead;
- PaymentOrder;
- EmailCampaign;
- EmailSubscriber;
- ChatSession;
- ChatMessage;

- Integration;
- GdprErasureRequest;
- CompanyErasureRequest;
- BackupSnapshot;
- AuditLog.

7.2. Relational Modeling

Relații:

- User 1:N Session;
- User N:M Company prin Membership;
- Company 1:N Document;
- Company 1:N AccountingPeriod;
- Company 1:N FinancialAnalysis;
- User 1:N Document uploads;
- User 1:N AuditLog;
- Company 1:N PaymentOrder.

7.3. Indexing

Tehnici:

- unique indexes;
- compound unique;
- foreign key indexes;
- lookup indexes.

Exemple:

```
User.email unique
Company.cui unique
Membership.userId+companyId unique
AccountingPeriod companyId+year+month unique
Session.tokenHash unique
PasswordResetToken.tokenHash unique
```

7.4. JSON Configuration Storage

Aplicare:

- site settings;
- report data;
- page sections;
- financial analysis details;
- encrypted integration configs.

Avantaje:

- flexibilitate;
- extensibilitate;

- compatibilitate cu module configurabile.

Riscuri:

- validare dificilă;
- migrare schemă JSON;
- query limitat;
- necesită versionare internă.

8. Tehnici de securitate

8.1. Authentication Security

Tehnici:

- password hashing cu bcrypt;
- session token hash;
- HttpOnly cookies;
- secure cookies pe HTTPS;
- SameSite;
- session expiration;
- session invalidation;
- generic auth error messages.

8.2. Authorization

Tehnici:

- RBAC;
- role guard;
- membership guard;
- admin guard;
- document ownership check;
- company assignment check.

8.3. Two-Factor Authentication

Tehnici:

- email OTP;
- code hash;
- expiration;
- max attempts;
- consumed token;
- session creation after verification.

8.4. Password Reset Security

Tehnici:

- token random;
- token hash;
- expiration configurable;
- single-use token;
- session invalidation after reset.

8.5. Email Change Security

Tehnici:

- current password validation;
- confirmation on old email;
- confirmation on new email;
- expiration window;
- session invalidation.

8.6. File Security

Tehnici:

- MIME allowlist;
- file size limit;
- storage outside direct public routing;
- authenticated download endpoint;
- membership-based authorization;
- path traversal guard.

8.7. HTTP Security Headers

Header-e:

- X-Content-Type-Options;
- X-Frame-Options;
- Referrer-Policy;
- Permissions-Policy;
- X-DNS-Prefetch-Control;
- Cross-Origin-Opener-Policy;
- X-Permitted-Cross-Domain-Policies;
- HSTS pe HTTPS.

8.8. Secret Management

Tehnici:

- environment variables;
- encrypted config storage;
- no secret logging;

- no credentials in repository;
- rotation policy;
- least privilege.

8.9. GDPR Security

Tehnici:

- data minimization;
- retention policy;
- pseudonymization/anonymization;
- access control;
- audit trail;
- backup purge workflow;
- consent tracking;
- DPIA/GAP reporting.

9. Tehnici de conformitate GDPR/DPIA/GAP

9.1. Data Mapping

Elemente:

- data source;
- data category;
- data subject;
- processing purpose;
- legal basis;
- storage location;
- retention period;
- recipient;
- transfer;
- deletion mechanism.

9.2. DPIA

Pași:

- processing description;
- necessity assessment;
- proportionality assessment;
- risk identification;
- risk scoring;
- mitigation controls;
- residual risk;

- approval;
- periodic review.

9.3. GAP Analysis

Categorii:

- privacy notice;
- consent;
- cookies;
- data subject rights;
- processors;
- transfers;
- security;
- retention;
- breach process;
- audit logging;
- access management.

9.4. Retention Engineering

Tehnici:

- retention rules;
- legal hold;
- account deactivation;
- delayed erasure;
- anonymization;
- purge marking;
- backup purge confirmation.

10. Tehnici SEO

10.1. Technical SEO

Tehnici:

- semantic HTML;
- metadata;
- canonical URL;
- sitemap;
- robots;
- hreflang;
- Open Graph;
- Twitter cards;

- structured content;
- image alt;
- performance optimization;
- mobile-friendly layout.

10.2. On-Page SEO

Tehnici:

- title optimization;
- meta description;
- H1-H6 hierarchy;
- internal linking;
- keyword mapping;
- content freshness;
- localized content;
- page-specific metadata.

10.3. Social SEO

Tehnici:

- Open Graph title;
- Open Graph description;
- Open Graph image;
- social share URLs;
- consistent branding;
- social profile linking.

10.4. Performance SEO

Tehnici:

- optimized images;
- compression;
- cache headers;
- JS reduction;
- CSS optimization;
- server-side rendering;
- lazy loading;
- Core Web Vitals monitoring.

11. Tehnici de testare

11.1. Static Testing

Tehnici:

- TypeScript check;
- linting;
- dependency audit;
- dead code scan;
- config validation.

11.2. Unit Testing

Candidați:

- financial metrics;
- economic metrics;
- SWOT generation;
- GDPR scoring;
- retention calculation;
- payment total calculation;
- email template rendering.

11.3. Integration Testing

Candidați:

- server action + DB;
- API route + DB;
- auth + session;
- document upload + download;
- payment order creation;
- chat message lifecycle.

11.4. E2E Testing

Fluxuri:

- login admin;
- create user;
- create company;
- assign membership;
- login client;
- upload document;
- generate analysis;
- reset password;
- run GDPR report;
- change template.

11.5. Visual Regression Testing

Tehnici:

- screenshot baseline;
- viewport matrix;
- pixel diff;
- threshold;
- cross-browser capture.

11.6. Smoke Testing

Rute:

- /;
- /admin;
- /portal;
- /autenticare;
- /gdpr;
- /sitemap.xml;
- /robots.txt.

11.7. Security Testing

Tehnici:

- auth bypass checks;
- RBAC checks;
- membership isolation;
- file traversal test;
- CSRF-sensitive flows review;
- header checks;
- session expiration;
- password reset token reuse.

11.8. Performance Testing

Tehnici:

- Lighthouse;
- WebPageTest;
- server response timing;
- DB query profiling;
- image size audit;
- bundle size analysis;
- caching verification.

12. Tehnici debugging

12.1. Reproducibility

Elemente:

- URL;
- role;
- user ID;
- company ID;
- browser;
- viewport;
- environment;
- timestamp;
- exact steps;
- expected result;
- actual result.

12.2. Root Cause Analysis

Tehnici:

- log inspection;
- DB inspection;
- network inspection;
- console inspection;
- binary search;
- config diff;
- regression diff;
- dependency diff;
- environment diff.

12.3. Fault Isolation

Straturi:

- browser;
- frontend component;
- server action;
- API route;
- auth/session;
- DB;
- external service;
- hosting;
- environment variables;
- file permissions.

12.4. Regression Control

Tehnici:

- reproduce before fix;
- minimal patch;
- retest same flow;
- test adjacent flows;
- update documentation;
- create checklist entry.

13. Tehnici DevOps și deployment

13.1. Build Strategy

Tehnică:

local prebuilt build

Motiv:

- hosting shared cu limită de procese;
- evitarea EAGAIN;
- evitarea fork Resource temporarily unavailable;
- evitarea build failure pe cPanel.

13.2. Deployment Strategy

Pași:

- backup;
- build local;
- package prebuilt;
- upload;
- install dependencies;
- Prisma generate;
- optional Prisma push;
- restart app;
- smoke test.

13.3. Runtime Strategy

Componente:

- app.js;
- server.js;
- Passenger/cPanel Node.js;
- .next;
- storage/runtime.log.

13.4. Rollback Strategy

Tehnici:

- păstrare arhivă precedentă;
- backup DB;
- restore files;
- restore DB;
- restart Node app;
- smoke test.

13.5. Environment Parity

Medii:

- local XAMPP;
- staging opțional;
- production cPanel.

Tehnici:

- .env controlat;
- același Node major version;
- același package manager;
- aceeași schemă DB;
- aceeași arhivă prebuilt.

14. Tehnici observability

14.1. Logging

Surse:

- storage/runtime.log;
- cPanel logs;
- browser console;
- network logs;
- DB errors;
- payment logs;
- email sending logs.

14.2. Metrics

Metrici recomandate:

- uptime;
- response time;
- error rate;
- login failures;
- password reset requests;

- email delivery status;
- chat sessions;
- payment order statuses;
- GDPR report score;
- document upload count.

14.3. Health Checks

Endpointuri recomandate:

- app health;
- DB health;
- storage health;
- auth health;
- external integration health.

14.4. Alerting

Evenimente:

- 503;
- DB connection failure;
- high error rate;
- failed email batch;
- failed payment callback;
- storage write failure;
- suspicious login attempts.

15. Tehnici de management configurații

15.1. Environment Configuration

Variabile:

- NODE_ENV;
- NEXT_PUBLIC_SITE_URL;
- DATABASE_URL;
- AUTH_SECRET;
- SMTP variables;
- seed variables.

15.2. Database Configuration

Tabel:

SiteSetting

Categorii:

- SEO;

- payments;
- email;
- chat;
- maintenance;
- security;
- integrations;
- compliance;
- template;
- easter egg.

15.3. Secure Configuration

Tehnici:

- encryption;
- masking;
- no plain output;
- no secret in manual screenshots;
- rotation support.

16. Tehnici de integrare externă

16.1. API Integration Pattern

Pași:

- credential storage;
- connection test;
- request signing/authentication;
- fetch;
- transform;
- validate;
- persist;
- log;
- retry;
- error handling;
- sync status.

16.2. Providers

Provideri țintă:

- Oblio;
- SmartBill;
- TDec;

- WinMentor;
- Nexus;
- Saga C;
- Saga PS;
- ANAF;
- Netopia;
- EuPlătesc;
- PayU;
- Mailchimp;
- WhatsApp Business API.

16.3. Idempotency

Tehnici:

- external ID mapping;
- hash payload;
- unique import key;
- retry-safe operations;
- duplicate detection.

16.4. Sync Logging

Câmpuri recomandate:

- provider;
- operation;
- startedAt;
- finishedAt;
- status;
- recordsIn;
- recordsCreated;
- recordsUpdated;
- recordsFailed;
- errorCode;
- errorMessage;
- payload reference.

17. Tehnici de date financiare

17.1. Financial Data Normalization

Tehnici:

- period normalization;

- currency normalization;
- decimal precision;
- missing value handling;
- imported/manual source flag;
- source provider traceability.

17.2. Indicator Calculation

Indicatori:

- revenue;
- expenses;
- profit;
- net margin;
- revenue growth;
- expense ratio;
- current ratio;
- debt ratio;
- return on assets;
- cash coverage;
- working capital;
- asset turnover;
- autonomy;
- receivables share;
- payables share;
- tax pressure;
- cash expense days.

17.3. SWOT Generation

Tehnici:

- threshold rules;
- trend analysis;
- risk rules;
- opportunity mapping;
- recommendation mapping.

17.4. Financial Report Persistence

Tehnică:

- store summary fields;
- store details JSON;
- keep generated analysis immutable;

- regenerate on demand.

18. Tehnici de email/campanii

18.1. Local SMTP Campaigning

Tehnici:

- subscriber table;
- consent flag;
- campaign draft;
- recipient generation;
- send loop;
- status tracking;
- failure logging.

18.2. Mailchimp Provider Switching

Tehnică:

```
if Mailchimp enabled → local disabled  
if Mailchimp disabled → local enabled
```

18.3. Deliverability

Tehnici recomandate:

- SPF;
- DKIM;
- DMARC;
- unsubscribe;
- bounce handling;
- complaint handling;
- sender reputation monitoring;
- rate limiting.

19. Tehnici de chat

19.1. Internal Chat Pattern

Entități:

- ChatSession;
- ChatMessage;
- settings;
- FAQ rules.

Flux:

- visitor opens widget;
- creates session;

- sends message;
- FAQ matcher;
- admin reply;
- optional fallback AI;
- close session.

19.2. FAQ AI Minimal Engine

Tehnici:

- keyword matching;
- normalized text;
- priority rules;
- fallback response;
- delay threshold.

19.3. Realtime Extension

Tehnici recomandate pentru viitor:

- polling;
- SSE;
- WebSocket;
- push notifications;
- operator assignment;
- unread counters.

20. Tehnici de plăți

20.1. Payment Order Pattern

Entitate:

PaymentOrder

Statusuri:

- created;
- pending payment;
- bank transfer pending;
- paid;
- failed;
- cancelled.

20.2. Gateway Integration Pattern

Pași:

- create order;
- sign payload;

- redirect;
- callback;
- verify signature;
- update status;
- generate invoice;
- audit log.

20.3. Manual Bank Transfer Pattern

Pași:

- create order;
- generate proforma;
- show bank details;
- admin confirms payment;
- mark paid;
- attach proof.

21. Tehnici de backup/restore

21.1. Backup Types

Tipuri:

- full file backup;
- database dump;
- user snapshot;
- company snapshot;
- storage backup;
- configuration backup.

21.2. Restore Types

Tipuri:

- full restore;
- partial user restore;
- partial company restore;
- file restore;
- DB restore;
- rollback deploy.

21.3. Backup Validation

Tehnici:

- checksum;
- restore test;

- schema compatibility;
- archive integrity;
- retention policy check.

22. Tehnici de versionare și control schimbări

22.1. Version Control

Tehnici:

- Git;
- feature branches;
- fix branches;
- tagged releases;
- commit messages;
- changelog.

22.2. Change Management

Flux:

- request;
- impact analysis;
- estimate;
- implementation;
- testing;
- documentation;
- release.

22.3. Release Management

Artefacte:

- release notes;
- build artifact;
- DB migration note;
- rollback plan;
- deployment checklist.

23. Tehnici de documentare

Tipuri:

- admin manual;
- user manual;
- developer manual;
- Scrum schema;

- Waterfall schema;
- debugging process;
- deployment checklist;
- API integration notes;
- GDPR/DPIA/GAP report;
- acceptance report.

Reguli:

- documentație versionată;
- actualizare după schimbări;
- separare operațional/tehnic;
- limitări explicit documentate.

24. Tehnici de control calitate

24.1. Quality Gates

Gate-uri:

- build pass;
- typecheck pass;
- DB generate pass;
- smoke routes pass;
- auth flow pass;
- admin flow pass;
- portal flow pass;
- responsive check pass;
- security checklist pass;
- documentation updated.

24.2. Acceptance Testing

Categorii:

- UAT admin;
- UAT client;
- UAT public site;
- UAT SEO/GDPR;
- UAT hosting.

24.3. Defect Lifecycle

Statusuri:

- new;
- triaged;

- assigned;
- in progress;
- fixed;
- retest;
- closed;
- deferred;
- rejected;
- cannot reproduce.

25. Tehnici de mentenanță

25.1. Corrective Maintenance

Aplicare:

- bug fixes;
- 503;
- layout broken;
- auth issues;
- DB issues.

25.2. Adaptive Maintenance

Aplicare:

- schimbări API;
- schimbări hosting;
- schimbări legislație;
- browser updates;
- dependency updates.

25.3. Perfective Maintenance

Aplicare:

- optimizare UX;
- optimizare performanță;
- funcții noi;
- rapoarte noi.

25.4. Preventive Maintenance

Aplicare:

- refactoring;
- dependency audit;
- security hardening;
- backup testing;

- monitoring.

26. Tehnici de refactoring

Tehnici:

- extract function;
- extract component;
- extract service;
- remove duplication;
- replace conditional with strategy;
- introduce DTO;
- isolate side effects;
- centralize config;
- normalize naming;
- split large actions.

Reguli:

- refactor separat de feature major;
- test înainte/după;
- păstrare comportament;
- documentare schimbare dacă afectează API/public behavior.

27. Tehnici de performanță

27.1. Frontend Performance

Tehnici:

- image optimization;
- lazy loading;
- CSS minimization;
- JS bundle control;
- server rendering;
- font optimization;
- caching static assets.

27.2. Backend Performance

Tehnici:

- query optimization;
- indexes;
- pagination;
- no over-fetching;

- streaming files;
- caching settings;
- async jobs pentru operații grele.

27.3. Database Performance

Tehnici:

- compound indexes;
- query explain;
- normalized relations;
- selective JSON use;
- batch operations;
- transaction boundaries.

28. Tehnici de compatibilitate

Browsere:

- Chrome;
- Edge;
- Firefox;
- Waterfox;
- Opera.

Device classes:

- desktop;
- laptop;
- tablet;
- smartphone;
- kiosk.

Tehnici:

- progressive enhancement;
- responsive layout;
- standard CSS;
- no browser-specific critical dependency;
- fallback UI;
- cross-browser smoke testing.

29. Tehnici de risc

29.1. Risk Identification

Categorii:

- hosting resource limits;
- payment correctness;
- GDPR compliance;
- data loss;
- API instability;
- email deliverability;
- unauthorized access;
- broken responsive layout.

29.2. Risk Mitigation

Tehnici:

- backup;
- staging;
- feature flags;
- maintenance mode;
- role guards;
- audit logs;
- rollback plan;
- manual confirmation for high-risk actions.

30. Matrice tehnici vs module

Modul	Tehnici principale
Site public	SSR, responsive, SEO, i18n, template strategy
Admin	RBAC, server actions, form validation, config object
Portal	auth guard, membership guard, document security
CMS	JSON sections, metadata, template binding
Documente	file validation, protected download, storage isolation
Analize	service layer, metric calculation, JSON report persistence
GDPR	DPIA/GAP, retention, anonymization, audit
Plăți	payment order, processor strategy, callback architecture
Email	provider strategy, campaign status tracking
Chat	session/message model, FAQ matcher, widget state
Integrări	credential encryption, provider strategy, sync logs
Deploy	prebuilt artifact, cPanel custom server, smoke testing

31. Standard tehnic recomandat

Pentru fiecare modificare:

Requirement

- Impact analysis
- Technical design
- Implementation
- Local test
- Regression test
- Documentation update
- Prebuilt build
- Deploy
- Smoke test
- Monitoring

Pentru fiecare bug:

Reproduce

- Isolate
- Fix minimal
- Retest
- Regression check
- Document

Pentru fiecare integrare externă:

Credential model

- Connection test
- Sandbox implementation
- Idempotent import
- Error handling
- Sync logs
- Production rollout

32. Concluzie tehnică

Setul complet de metode și tehnici aplicabile aplicației include:

- SDLC hybrid;
- MVC-like architecture;
- component-based frontend;
- server actions;
- API routes;
- Prisma ORM;
- relational data modeling;
- strategy/factory/guard/service patterns;
- RBAC;
- 2FA;
- token security;
- protected file serving;
- GDPR retention;
- DPIA/GAP;
- SEO technical/on-page/social;
- automated/manual testing;
- debugging workflow;
- prebuilt deployment;
- observability;

- backup/restore;
- risk management;
- quality gates;
- maintenance lifecycle.