Capstone Project II

Contractor Connect

System Implementation 1

The Four Group

Presented by:

Adel Alhajhussain – Backend Developer

Mehdi Jazi – Database Management

Shifa Karnelia – UI Developer

Sana Karnelia – Frontend Developer

Eric Laudrum – Quality Assurance

Instructor: Prof. Laily Ajellu

Project Description

ContractorConnect is a web-based platform designed to connect **homeowners** with **contractors** for residential projects through a structured and transparent system.

Project Idea:

The system centralizes how projects are posted, discovered, and managed, replacing informal communication and fragmented hiring methods.

What the Project Achieves:

- Enables homeowners to easily find and evaluate contractors
- Allows contractors to discover and bid on relevant projects
- Provides a role-based dashboard experience (Admin / Homeowner / Contractor)
- Improves transparency, organization, and workflow efficiency
- Creates a scalable digital solution for project management and service matching

Technology Used

Backend / Server-Side
- PHP
- CodeIgniter 4 (CI4)
- MVC Architecture

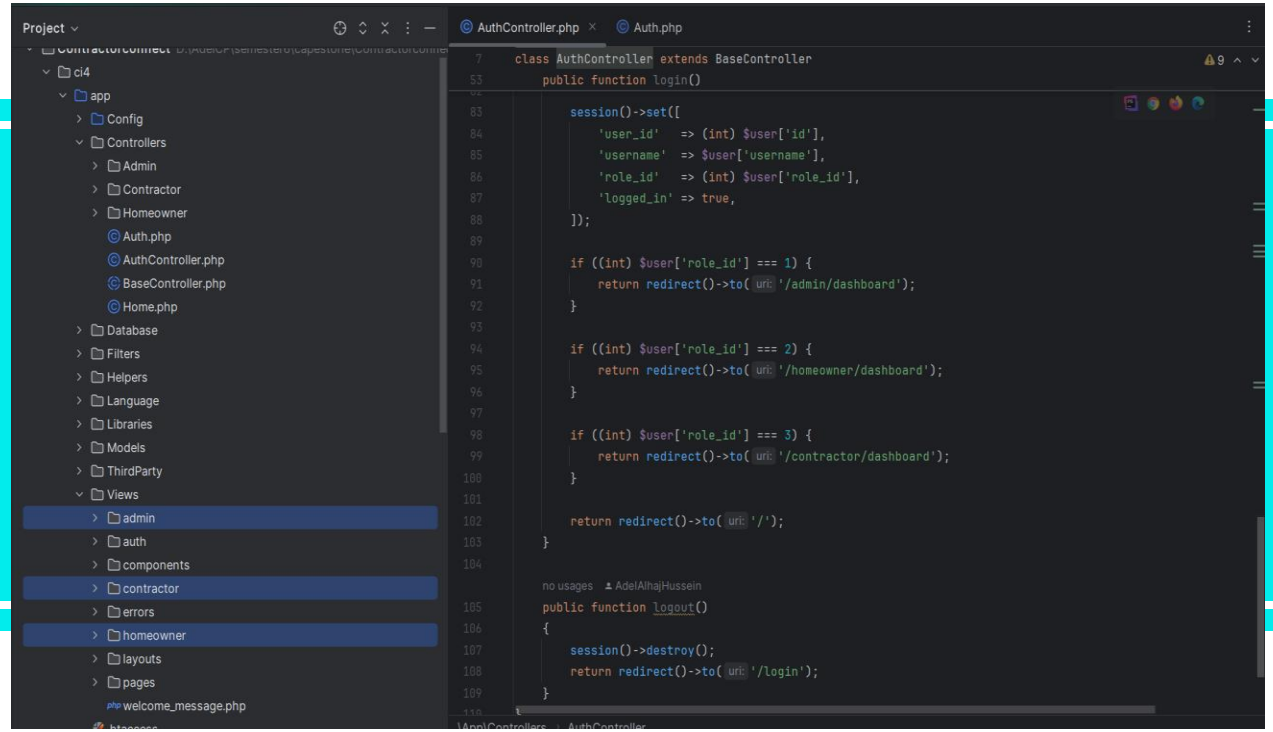
Database
- MySQL

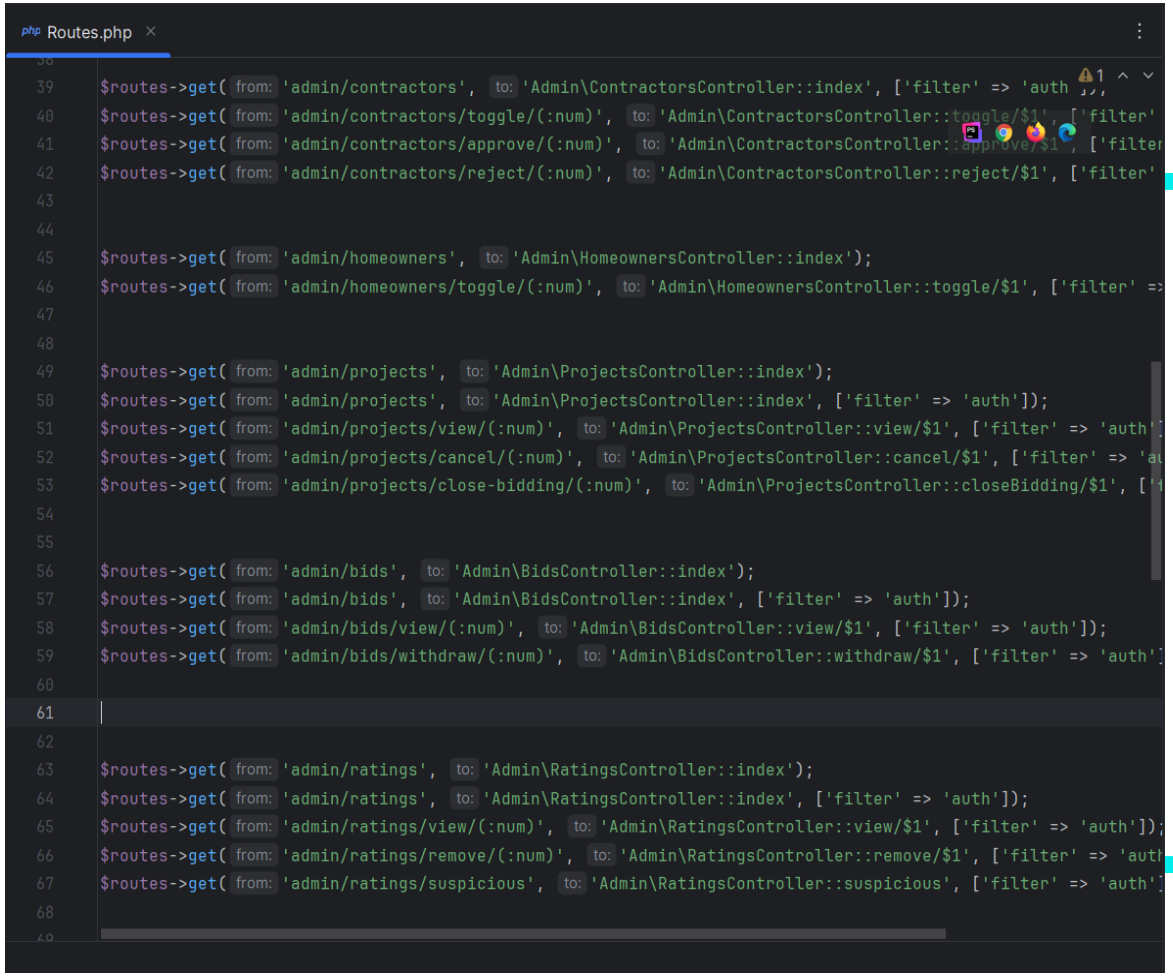
Frontend
- HTML
- CSS
- Bootstrap

Version Control
- Git
- GitHub

Role + CI4 Backend Architecture

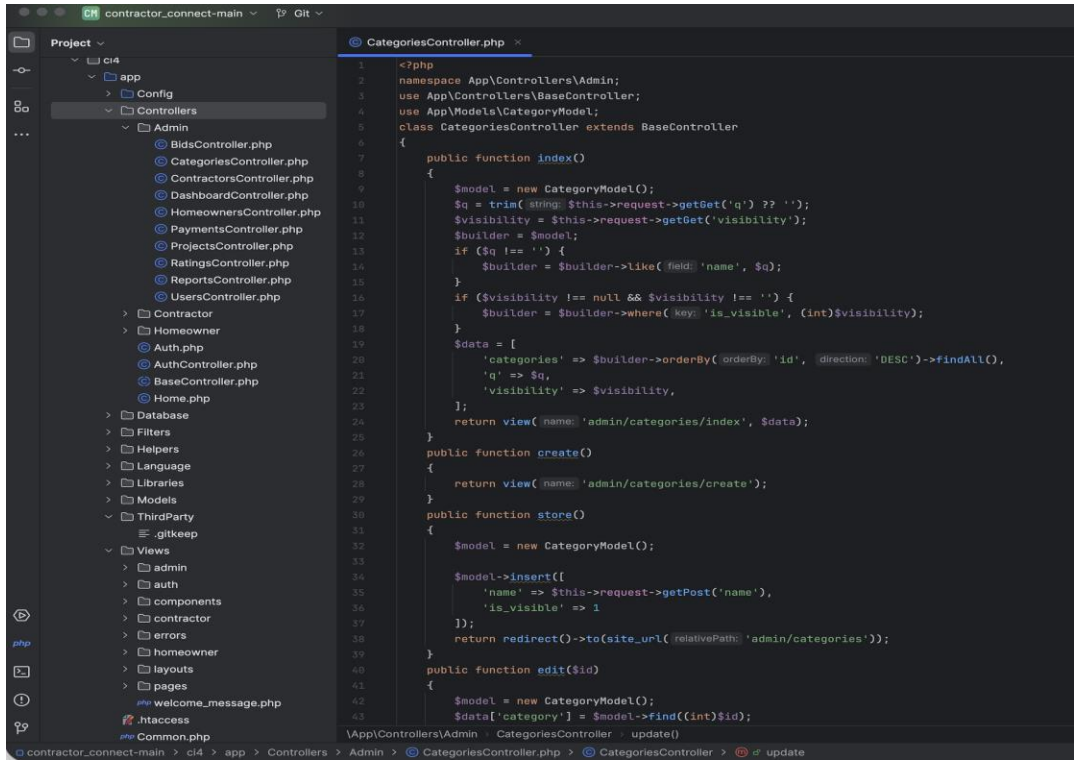
- ❖Backend implemented using CodeIgniter 4 (CI4)
- ❖Controllers organized by module: Admin / Contractor / Homeowner
- ❖Central AuthController for authentication (register/login)
- ❖Clean structure supports maintainability and easier future features

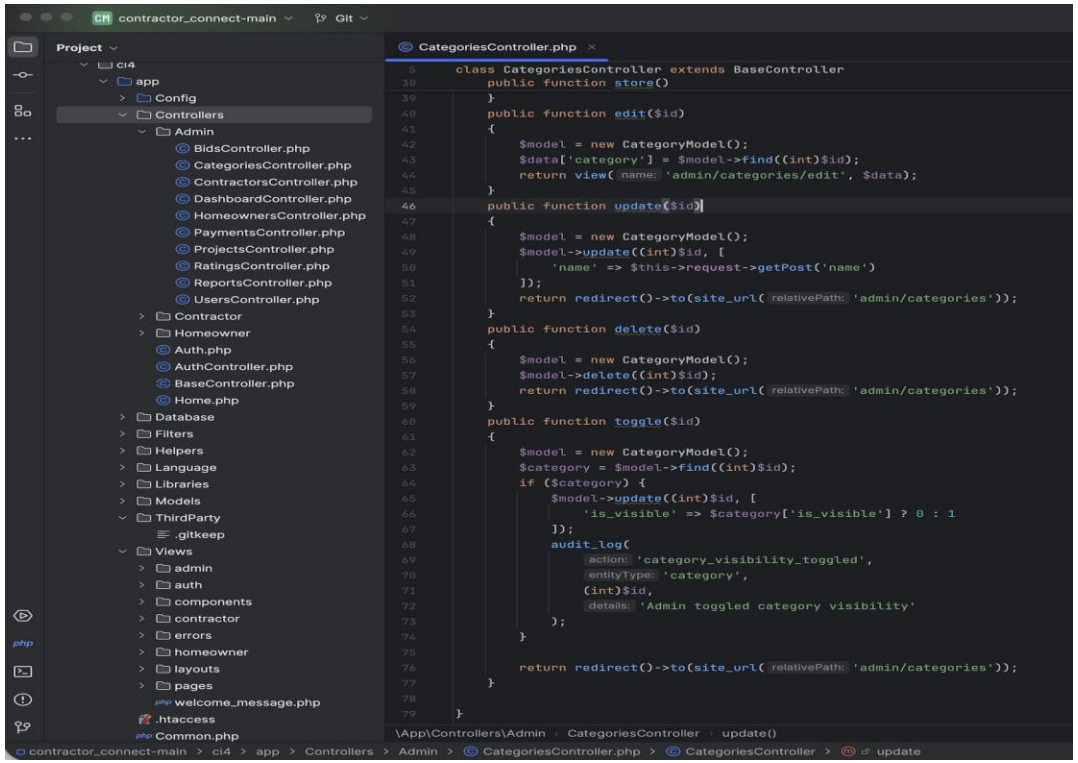




Routing Setup

- All routes are defined in app/Config/Routes.php
- Routes map URLs → controller methods (login, logout, admin pages)
- Dynamic parameters supported (e.g., (:num) for IDs)
- Auth filter protects admin routes from unauthorized access

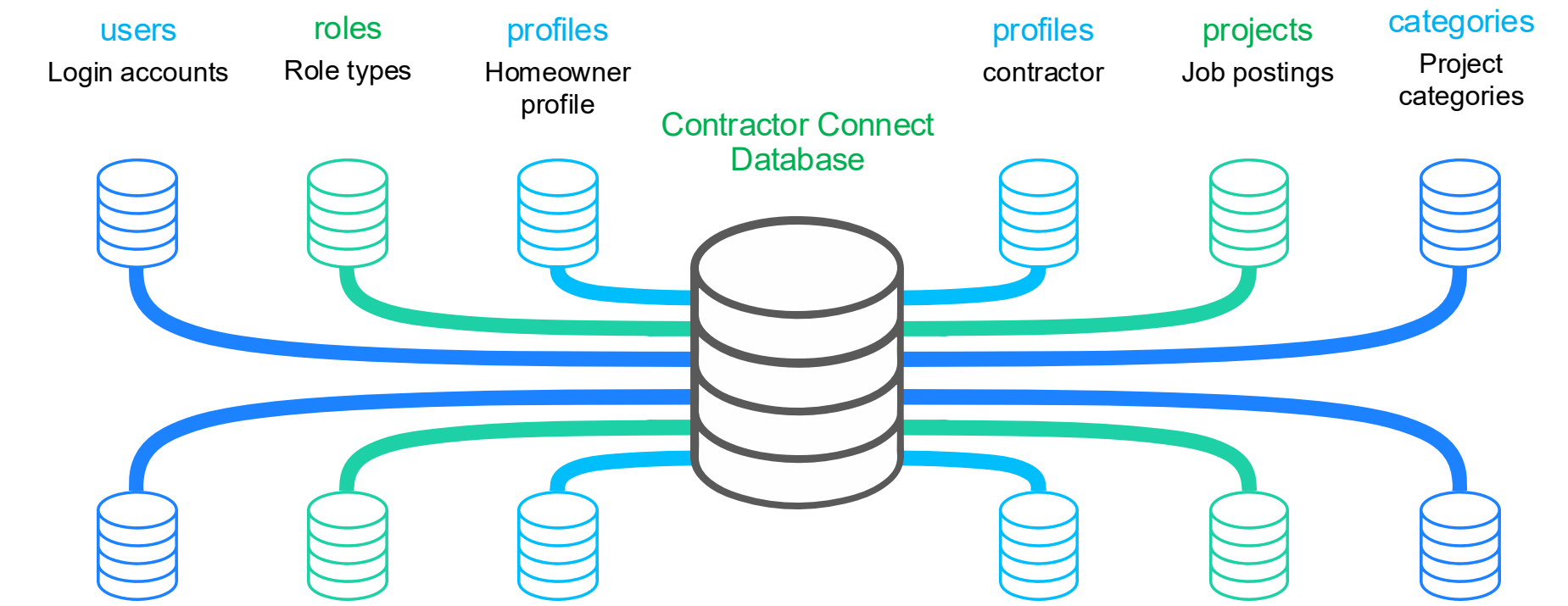




CRUD Implementation

- CategoriesController demonstrates full CRUD workflow
- Create: insert() adds new category records
- Read: findAll() loads category list (with filters/search)
- Update: update(id, data) edits category details or visibility
- Delete: delete(id) removes records when needed

Database Schema Overview



ERD Overview
(Entities & Relationships)

- Core entities: users, contractor_profiles, projects, bids

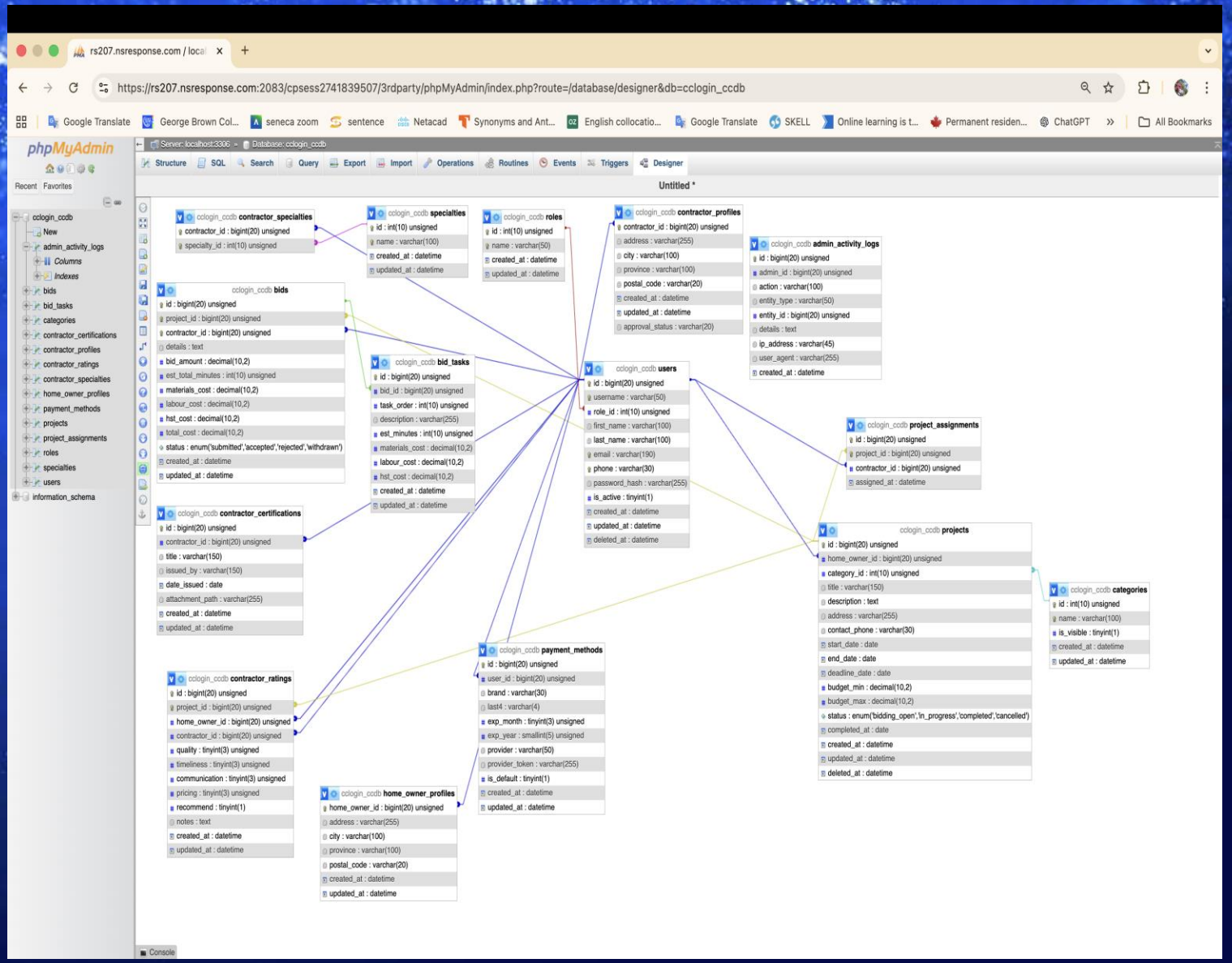
- users → contractor_profiles (1:1) contractor account details

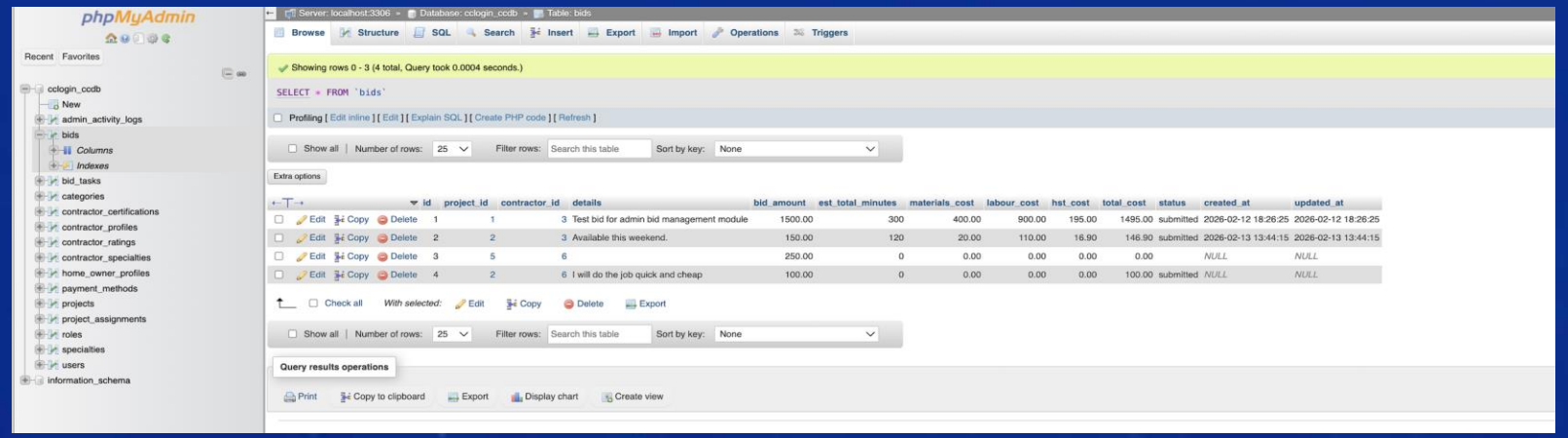
- users → projects (1:many) homeowner posts projects

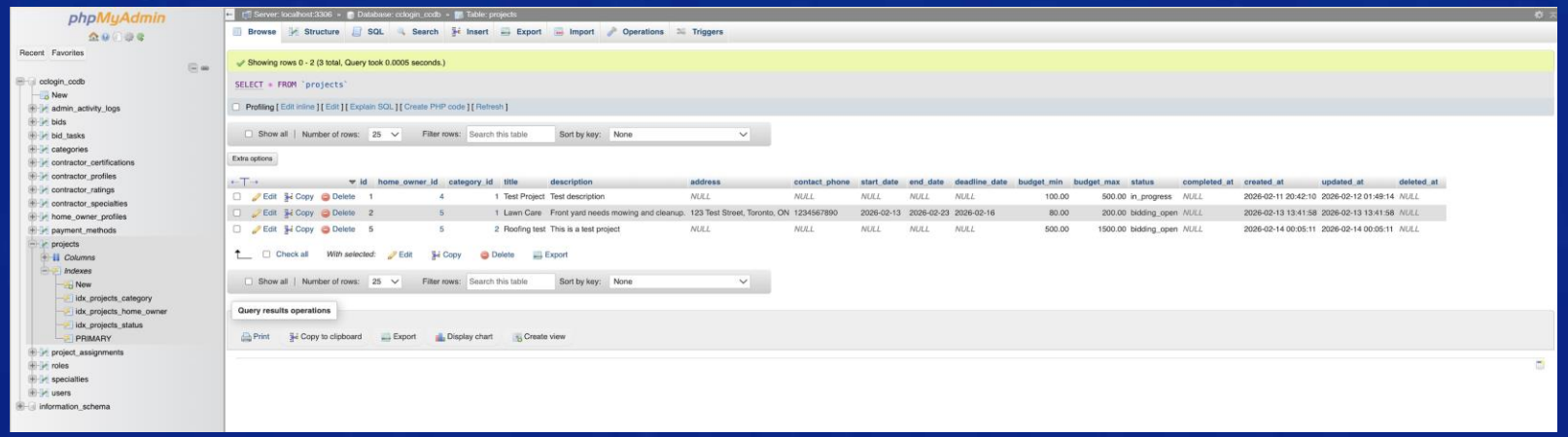
- projects → bids (1:many) each project receives bids

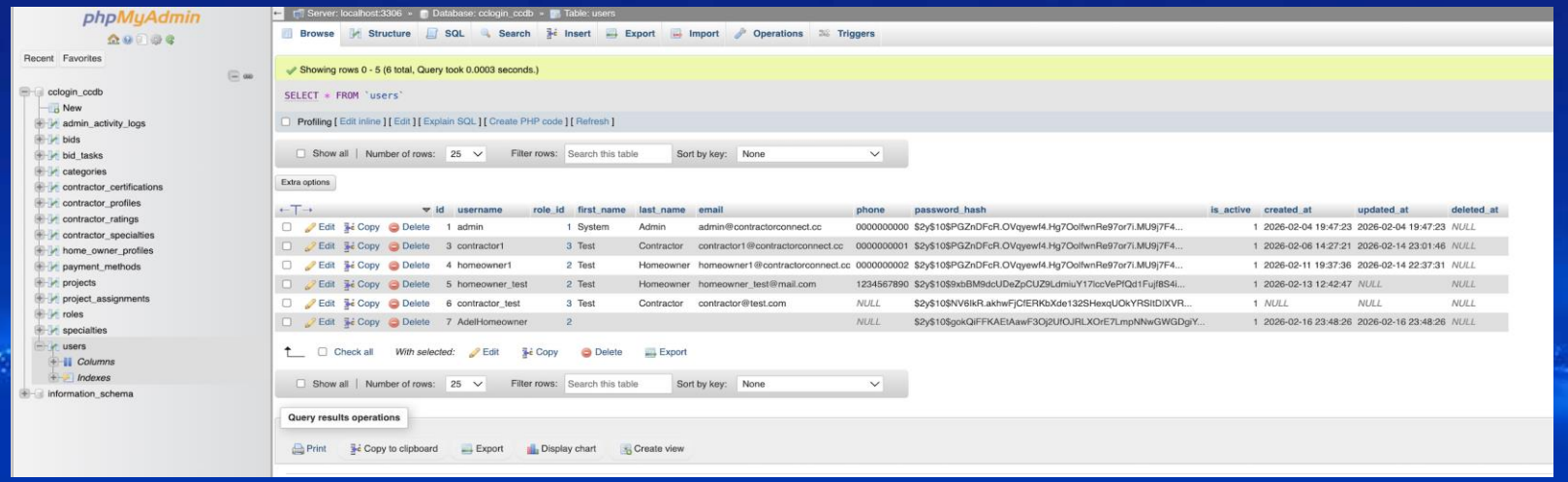
- contractor_profiles → bids (1:many) contractor submits multiple bids

- Foreign keys ensure data integrity (user_id, project_id, contractor**_id**)



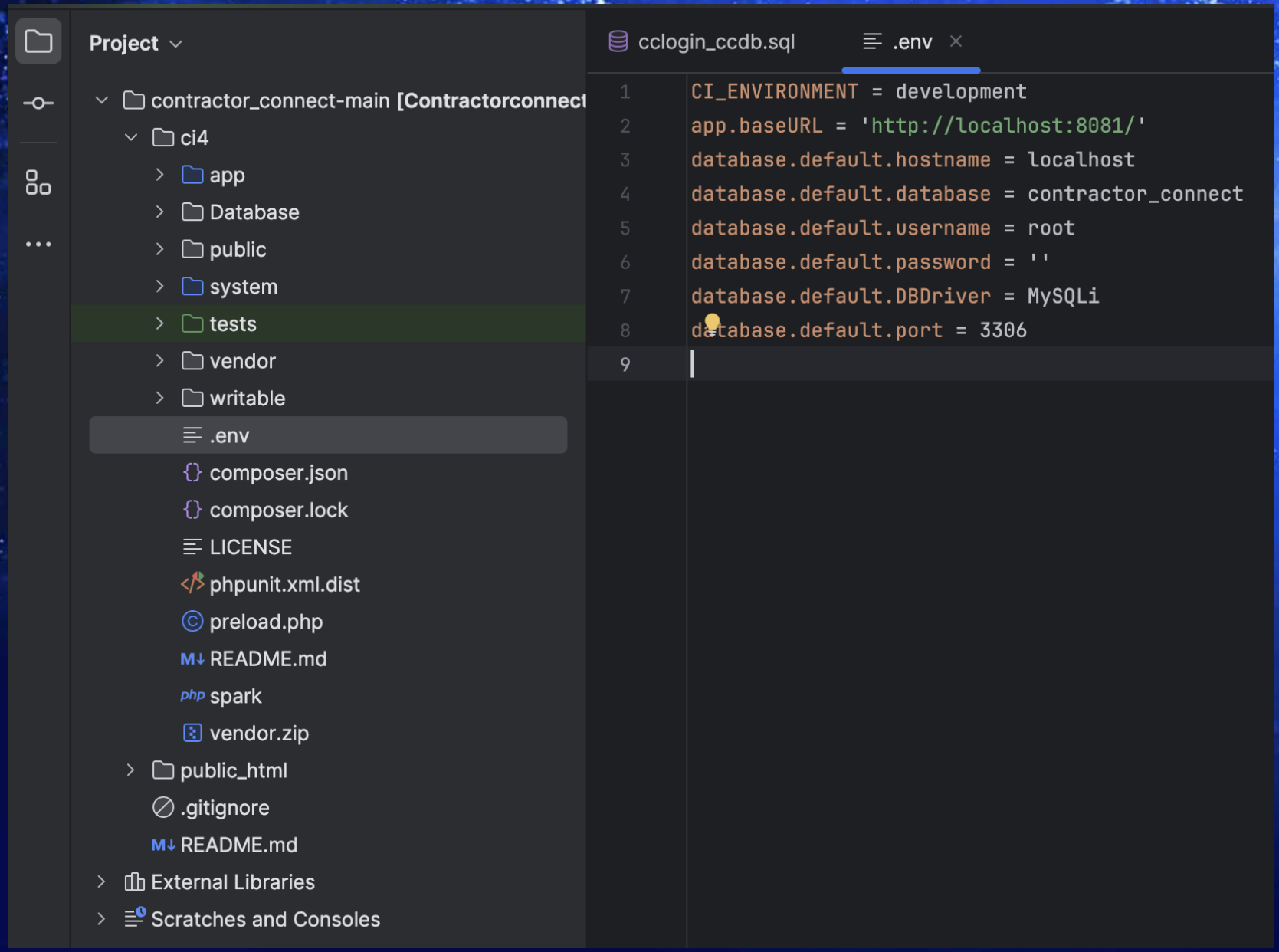






Main Tables & Sample Records (Users • Projects • Bids)

- Core tables used in the prototype: users, projects, bids
- users stores login + roles (admin / homeowner / contractor)
- projects created by homeowner (home_owner_id → users.id)
- bids submitted by contractor (contractor_id → users.id, project_id → projects.id)
- Sample data proves full flow: user → post project → submit bid
- Linked with primary keys + foreign keys to keep data consistent

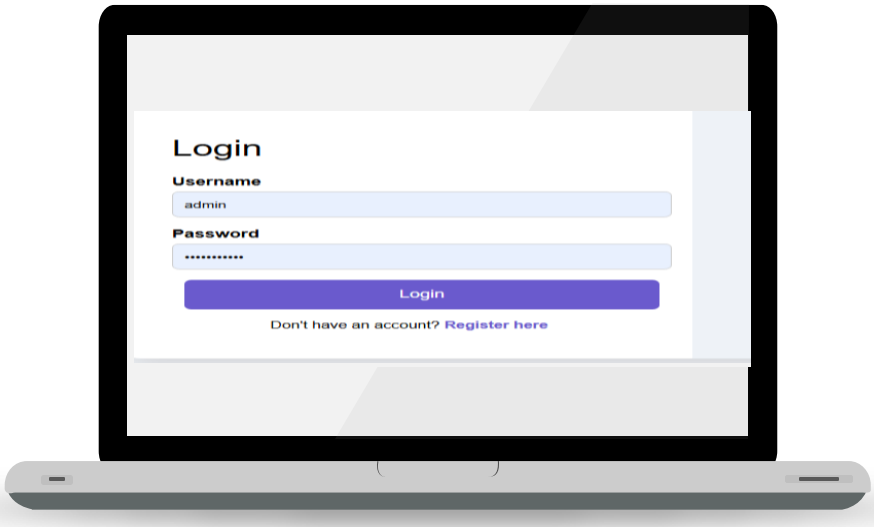


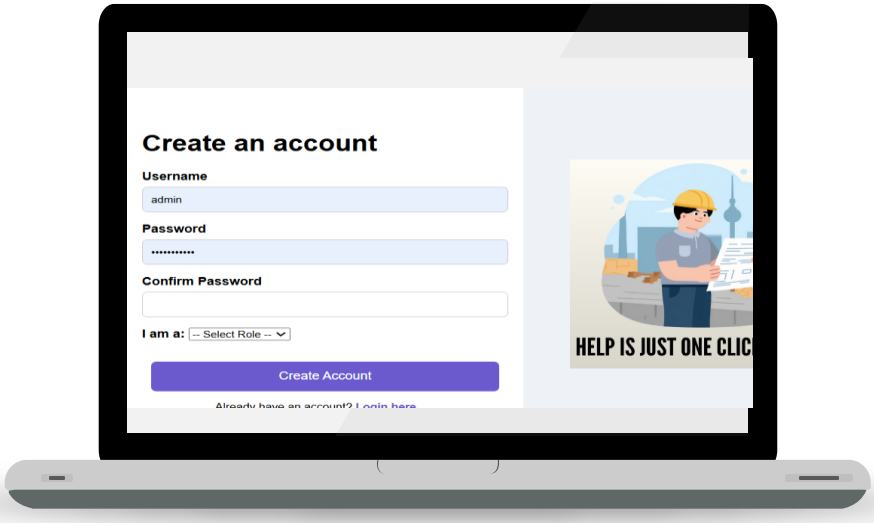
CI4 Database Connection Deliverables + Next Steps / Risks

- DB config in .env
- CI4 connects to MySQL
- Models: User / Project / Bid
- Flow works: project → bid → records
- Done: ERD + tables + sample data
- Next: indexes + validation + FK rules

Frontend Architecture & UI Design

- Designed user interface for all pages
- Implemented responsive layouts using HTML & CSS
- Created dashboard UI components
- Improved navigation and user experience
- Integrated frontend views with backend controllers

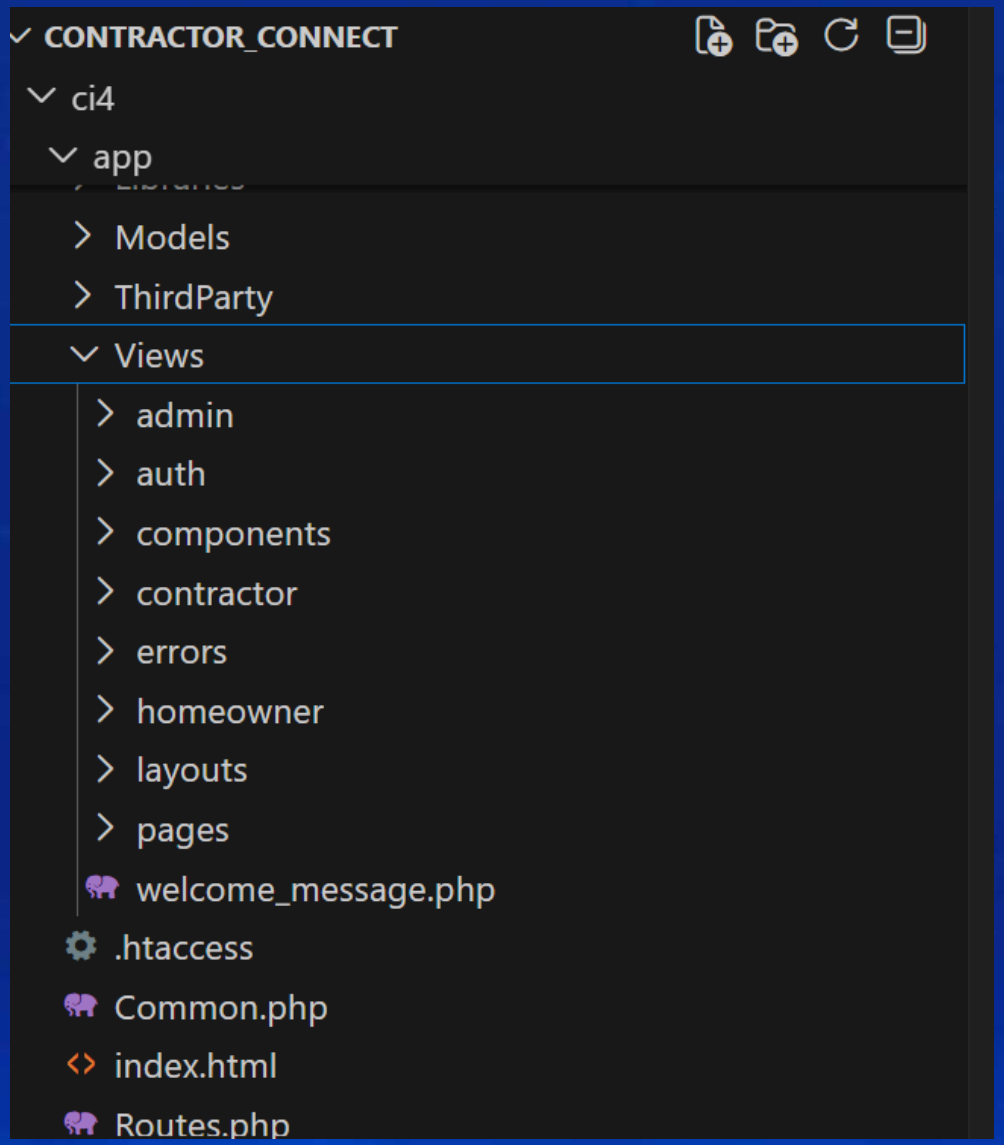


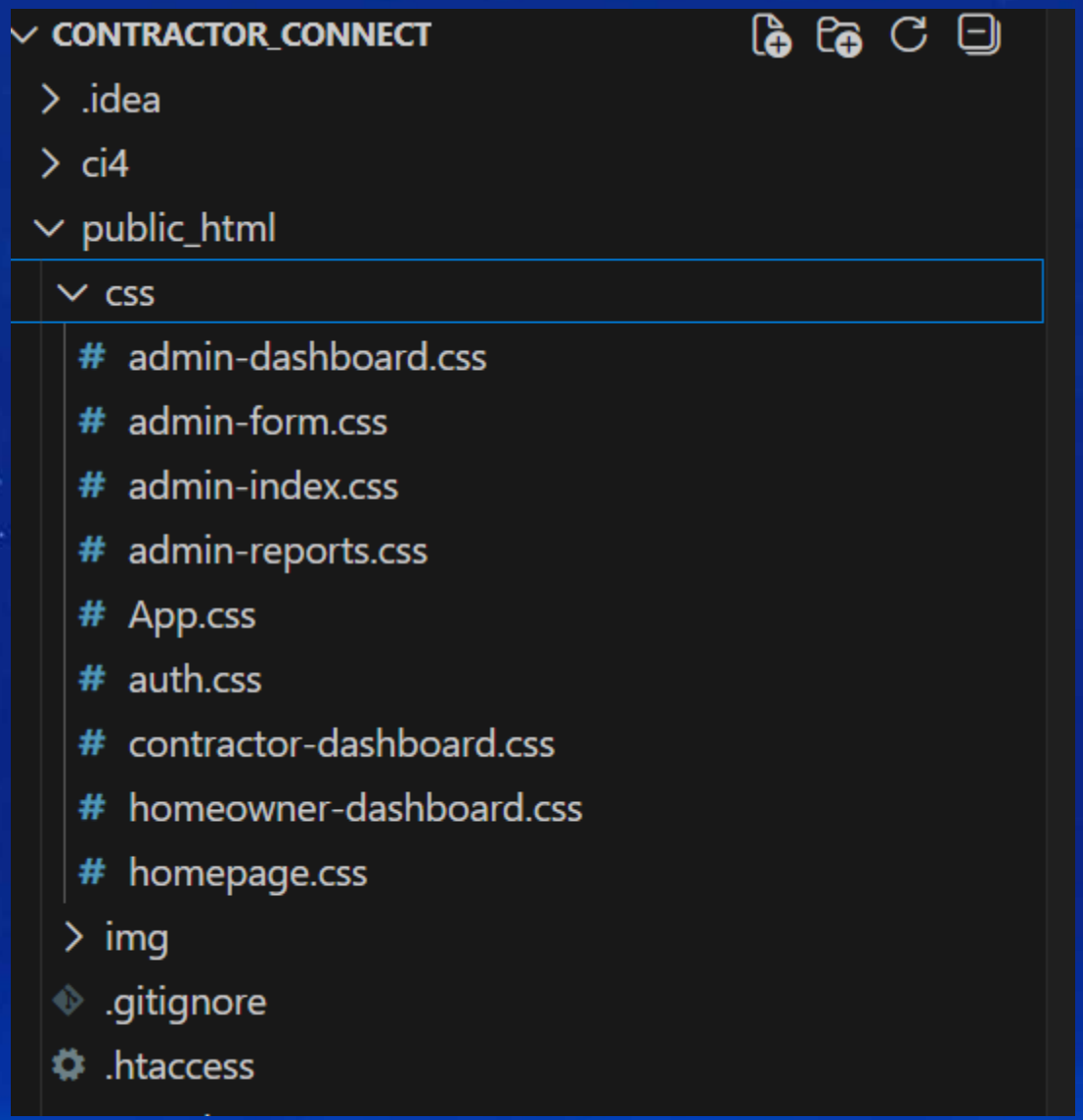


Frontend Architecture & UI Design

Technologies Used

- HTML5
- CSS3
- PHP (Views in CodeIgniter)
- MVC Architecture
- Bootstrap





Folder Structure

Planned Improvements :-

- Fix issues in the **User Registration process**
- Add **“Back to Dashboard” navigation button** in all role-based views (Admin, Homeowner, Contractor)
- Implement **Edit Project functionality** for both Contractors and Homeowners
- Enable **View All Project Bids** for Contractors and Homeowners
- Fix and improve **View & Update Profile** functionality for both roles
- Enhance **Contractor Dashboard** by adding more management features
- Adding a consistent footer across all pages for better structure and navigation

Technical Challenges:-

- Adding new features may impact existing APIs and cause:
  - Page loading failures
  - Routing issues
  - Data binding errors
- Maintaining stability while extending features requires:
  - Proper testing before deployment
  - Careful integration between frontend and backend
  - Avoiding breaking existing controller logic

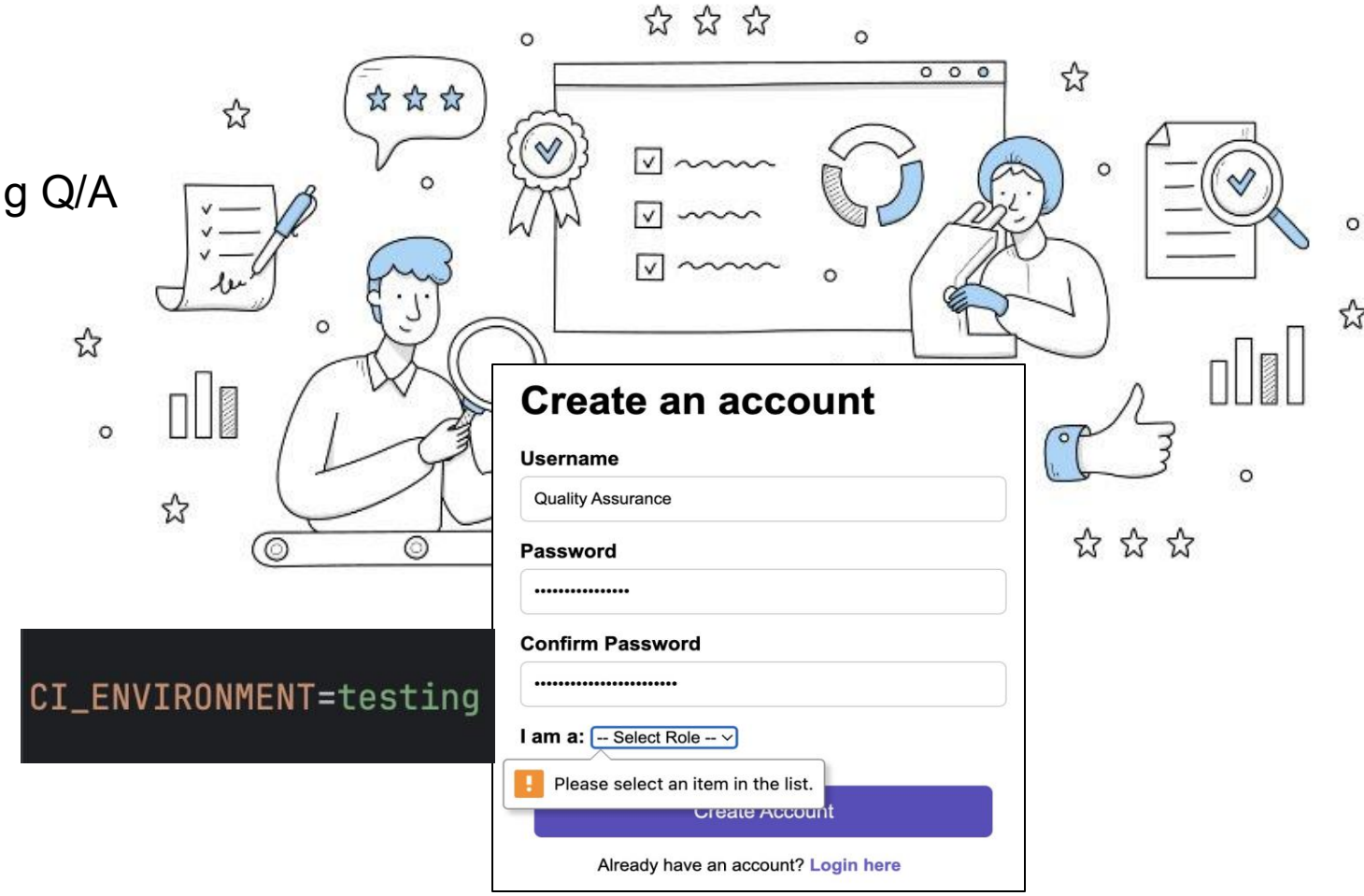
Quality Assurance

Learning Plan:
- LinkedIn Learning:
  - Programming Foundations: Software Testing Q/A

Testing Framework:
- CodeIgniter 4 Testing Suite
  - PHPUnit

State Management:
- DatabaseTestTrait
- Local MySQL

Data Strategy:
- FakerFactory



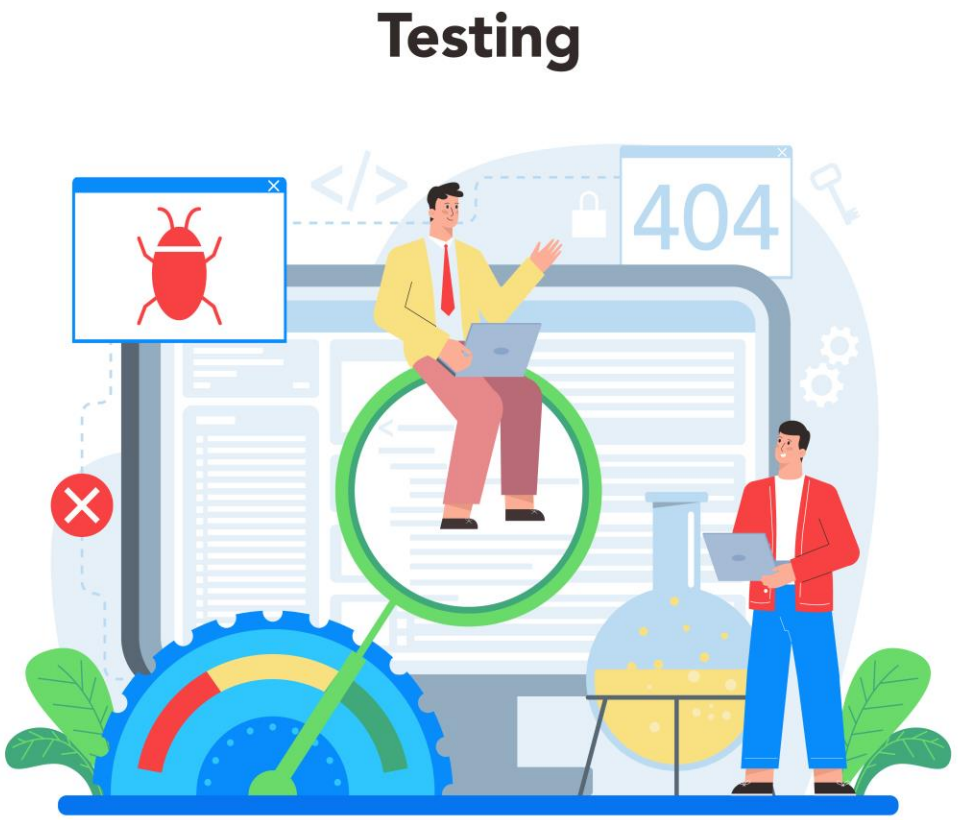
Unit Testing

Integration Testing:
- Testing full response cycle

Model Testing (Business logic):
- Project creation fails without a title
- Verify deadline has not expired
- Budget minimum can't exceed maximum

Feature Testing (User's path):
- Is the user routed to the appropriate page on login?

- Ex:
  - Simulating user login with POST request
  - Verifying redirects based on roles
  - Verifying Auth filters are protecting restricted routes



Contractor Connect

Challenges:
- Normalizing URLs to handle prefixes
- Using Query Builder to isolate authentication logic
- Working with current code

Next Steps:
- Connect GitHub directly to cPanel
- Integrating GitHub Actions
- Install a Driver (Xdebug) to identify missed edge cases

Learning Plan:
- LinkedIn Learning Course: GitHub Actions for CI/CD
- Read documentation on CI4 Testing Suite

Contractor Connect

Thank you

Adel Alhajhussain – Backend Developer
Mehdi Jazi – Database Management
Shifa Karnelia – UI Developer
Sena Karnelia – Frontend Developer
Eric Laudrum – Quality Assurance