

INSTRUCTIONS TO THE COMPETITOR

Module B – Dynamic website with server-side rendering

Goal

Build a **server-side rendered (SSR)** web app.

You can optionally use JavaScript to enhance the user experience, but the initial content of each must be rendered on the server. Test to check: Fetch the page in a CLI tool (like `curl` or `wget`) and verify that the HTML contains the content.

Tech Rules

- Must work in Google Chrome.
- Use any CSS framework or your own styles, but keep branding consistent.

Website & Branding Integration

- Optional merge with Module A: You may embed your static pages from Module A into this app, or keep them separate.
- Consistent look: Colours, fonts, and logo placements must match across all pages (static & dynamic).
- Easy navigation: Put clear links in the main nav bar (or landing page buttons) so experts can reach Investors, Tours, Admin pages in ≤ 2 clicks.
- Media: Re-use images/media from Module A or create your own files.
- Admin section: Has a clear and polished design, it is very clear to the user how to approve / reject requests.

Global Requirements

- Authentication:
 - o To log in, users enter their email. The system sends a login link to their email. Clicking the link logs them in. Users are identified by their email. There are no passwords. The same is true for the admin user.
- Security:
 - o Implement a CSRF token on every form or action (POST/PUT/DELETE), usually with a hidden input field.
 - o Escape/encode user input in views to avoid cross-site scripting (XSS) attacks.
 - o Use parameterised queries or rely on an ORM to prevent SQL injection.
- Organization:
 - o Your code must be split into clear modules and ensure consistent separation of concerns (database, render / model, logic).
 - o Forms with missing or invalid data must show an error message. The error message must be clear, understandable and user-friendly.

Email

At various points, the system sends emails to users. You must not actually send emails, but mock their sending.

You can define the email body in plain text or HTML format.

Store the sent emails in a database table and display them on a dedicated page `/mock-emails`. The table must include the recipient, subject, body, and sent timestamp. Sent emails are sorted by their timestamp. The body of the email must be rendered with new lines preserved and links clickable. If you decided to use plain text emails, you must convert them to HTML for display on the `/mock-emails` page.

Additional requirements:

- The page must be accessible without authentication.
- Styling is not required, but the table must be readable.
- The page does not have to be linked anywhere. When in doubt you can specify the link in the README.md file of your Git repository.

Pages and Features

Each page must be accessible via a unique URL. The URLs must be descriptive and follow a consistent pattern.

Presenting Sponsor Page

The presenting sponsor page displays the logo of the approved presenting sponsors.

The page shows at least 3 sponsors. However, if there are fewer than 3 approved sponsors, the page replaces those with a placeholder image. The page is not limited to 3 sponsors, but must show at least 3 (real or placeholder).

Investors Page

The investors page allows website visitors to invest and show existing investments.

Total support amount: Displays the total amount of support in DKK by all investors across all investment types.

Turbine spots visual: Shows available, pending, and approved turbine spots. A turbine has **10 spots** and they are numbered from 1 to 10. Each spot can be funded by a single investor. Any website visitors can see the spots and their availability.

Investing: A link or button opens a form to submit an investment request:

- Requires name, email, address, phone number
- Choose type of investment which shows different fields:
 - o Fund turbine: Requires a turbine spot selection, a logo upload (PNG, max 1 MB) or a ≤25 char text.
 - o Presenting sponsor: Requires a logo upload (PNG, max 1 MB).
 - o Support amount: Requires a money field for the amount. (Currency is DKK)

After submitting a valid form, the user receives an email confirming their requests and that it is now pending.

- Subject: "Investment Request Confirmation #{investmentReference}"

- `{investmentReference}` is a unique reference generated by the system. The format is `INV-XXXXXXX`, where `XXXXXXX` is composed of 8 digits, incrementing by one for each investment request.

- Body:

Hi {name}!

Thank you for your investment request.

We have received your request and it is now pending approval.

Investment reference: {investmentReference}

Investment Type: {investmentType}

Turbine Spot: {turbineSpot} (if applicable)

Logo/Text: {logoOrText} (if applicable)

Support Amount: {supportAmount} (if applicable)

We will shortly review your request and provide you with details on the next steps (such as payment details).

Stay breezy and sustainable!

Additionally, if they chose to fund a turbine, the turbine spot is blocked for other users until the admin approves or rejects the request. The turbine spot shows as "pending".

Once an investment is approved by an admin:

- Fund turbines: The turbine spot shows the uploaded logo or text (and no longer shows "pending").
- Presenting sponsor: The logo shows on the presenting sponsor page.
- Support amount: The amount is added to the total support amount.

... and the user receives an email confirming the approval:

- Subject: "Investment Request Approved #{investmentReference}"
- Body:

Hi {name}!

Your investment request has been approved!

Investment reference: {investmentReference}

Stay breezy and sustainable!

If the admin rejects the investment request, the user receives an email with the rejection.

You must use the same subject and body as for the approval, but replace the word "Approved" with "Rejected".

Visitor Tours Page

The tours page allows users to book seats on tours.

- Shows all tours with date, time, and available seats left.
- Authenticated users can book seats on a tour. If the user books more seats than available, the system prevents over-booking and shows an error message, and allows the user to select fewer seats.

The booking form requires tour selection, number of seats, and personal details: name, address, and phone number.

The booking is linked to the authenticated user. A single user can book multiple tours, including multiple times on the same tour.

Once the booking form is submitted and successfully validated, the user is redirected to the Tours page, where they can see their bookings at the top of the page. The booking status is "confirmed".

Also after a successful booking, the system sends an email to the user with the booking details:

- Subject: "Wind Farm Tour Booking Confirmation"
- Body:

Hi {name}!

You have successfully booked {numberOfSeats} seats on our Wind Farm tour.

Date: {tourDate} {tourTime}

Booking Details: {linkToToursPage}

We look forward to seeing you.

Stay breezy and sustainable!

Once the booking is confirmed, the user can cancel it. They can do this by clicking a "Cancel" button next to the booking on the Tours page. The booking status changes to "cancelled". The user can cancel a booking at any time.

The seats booked by the user are released back to the available seats for that tour.

Admin /admin/* Pages

All admin pages are protected by authentication. Only users with the **admin** role can access them.

All admin pages should be accessible via the /admin/* URL pattern.

The route /admin shows the admin dashboard which contains links to the following pages:

- Investment Admin Page
- Tours Admin Page

Investment Admin Page

The investment admin page allows the admin to manage all investments made by users.

It shows all investments made by users, including the type, status, and personal details (name, email, address, phone number).

The admin can approve or reject the pending investment requests with a button.

Tours Admin Page

The tours admin page allows the admin to manage tours.

- Shows all tours with date, time, capacity and available seats.
- Allows the admin to create new tours.
- Tours without any bookings can be deleted.
- Each tour has a link to view bookings for that tour. Each booking shows the personal details and the number of seats booked.
- The admin can cancel individual bookings:
 - o releases the booked seats back to the available seats.
 - o if all seats are cancelled, the tour can be deleted.
 - o sends an email to the user with the cancellation details:
 - Subject: "Wind Farm Tour Booking Cancellation"
 - Body:

Hi {name}!

Your booking for the Wind Farm tour on {tourDate} at {tourTime} has been cancelled.

We apologize for any inconvenience this may cause.

If you have any questions, please contact us.

Stay breezy and sustainable!

Database

You are free to design the database schema as you see fit.

You must provide a SQL dump file with both the **structure** and **initial data** to seed the database.

It must be committed to the Git repository in the root directory as `seed.sql`.

The SQL dump must include the following data:

- 2 users:
 - o Email: `admin@localhost` / Role: admin
 - o Email: `user@localhost` / Role: user

- 10 turbines (unique names, e.g. Turbine 1, Turbine 2, etc.)
- 2 investments: Fund turbine: Turbine, text
- 2 investments: Fund turbine: Turbine, logo
- 2 investments: Presenting sponsor: Logo
- 5 tours: unique date + time, capacity 10
- 2 bookings: on same tour, 2 seats each

The SQL dump will be assessed for correctness and completeness.

You may include additional seed data as long as the required rows above remain intact.

Example user flow

These user flows demonstrate the expected functionality of the application. You can use it to test your implementation.

It does not cover all edge cases, but it should give you a good idea of how the application should work.

Tour booking flow

1. Alex opens the website, goes to the tours page, and sees all tours.
2. Alex signs up with their email address. Receives an email (found in /mock-emails), clicks the link to log in.
3. Alex selects a tour with available seats and books **2 seats**.
4. Alex receives a confirmation email with booking details.
5. Alex goes to the tours page and sees their booking at the top with status "confirmed".
6. Alex decides to cancel the booking.
7. Alex clicks the "Cancel" button next to the booking.
8. Alex receives an email confirming the cancellation.
9. Alex goes to the tours page and sees their booking status changed to "cancelled".
10. Alex can book the same tour again or a different one. The seats are available again.

Investment flow

1. Alex opens the website, goes to the investors page, and sees available turbine spots.
2. Alex signs up with their email address. Receives an email (found in /mock-emails), clicks the link to log in.
3. Opens form to invest, select "Fund turbine", enters personal details and adds text or uploads logo.
4. An admin approves the investment request.
5. Alex sees the turbine spot with their logo or text on the investors page.

More flows exist, but this should give you a good idea of the expected functionality.