

React + Ts

Forms handling
react-hook-form
JOI

התנהלות מול טפסים בריאקט

```
const [formData, setFormData] = useState({
  email: "",
  password: "",
});
const [formError, setFormError] = useState({
  email: "",
  password: "",
});
const [allowSubmit, setAllowSubmit] = useState(false);
```

- במה מסייע לנו ה-hook המיוחד useState בהקשר של טפסים בריאקט?
- מדוע שנרצה שערכי הטופס ישמרו ב-state?
- מה היא ולידציה?
- מדוע שנרצה ששגיאות ולידציה ישמרו ב-state?

```
const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {
  setFormData({
    ...formData,
    [e.target.name]: e.target.value,
  });
  setAllowSubmit(chckErrors());
};
```

Regular expressions

- מה הוא Regex?
- לשם מה נשתמש במתודה test?
- כיצד נוכל לבצע ולידציות על השדות בכל הקלדה של המשתמש?
- כיצד נוכל להציג הודעה של שגיאת ולידציה למשתמש?
- כיצד נוכל לוודא האם כל הטופס ולידי בכל הקלדה של המשתמש ולאפשר את כפתור השליחה בהתאם?

```
const chckErrors = () => {
  const passwordRegex = /((?=.*[0-9]{1})(?=.*[A-Z]{1})(?=.*[a-z]{1})(?=.*[!@#$$%^&*~]{1}).{8,20})/;

  const isEmailValid = formData.email.includes("@") && formData.email.includes(".");
  const isPasswordValid = passwordRegex.test(formData.password);

  setFormError({
    ...formError,
    email: !formData.email || !isEmailValid ?
      "Email is required and must be valid":
      "",
    password: !formData.password || !isPasswordValid ?
      "Password is required and must be valid":
      "",
  });

  return isEmailValid && isPasswordValid;
};
```

טפסים – תרגול – חלק א'

- צרו קומפוננטה חדשה בשם Login בהתאם לנספח.
- הקומפוננטה תציג טופס התחברות הכולל שדות אימייל ושדה סיסמא.
- טפלו בטופס כך שהשדות יעברו ולידציה בכל הקלדת משתמש.
- במידה ויש שגיאת ולידציה, הודעת השגיאה תופיע מתחת לשדה המבוקש.
- אפשרו גישה לכפתור השליחה רק במידה והטופס ולידי.
- במידה והטופס ולידי, הדפיסו את תוכנו לקונסול בעת לחיצה על הכפתור.

```
<form
  className="flex flex-col gap-4 rounded-lg p-4 shadow-lg"
  onSubmit={handleSubmit}
>
  <h1 className="text-2xl font-bold text-gray-800">Sign In</h1>
  <FloatingLabel
    type="email"
    name="email"
    variant="outlined"
    label="Email"
    onChange={handleChange}
  />
  <span className="text-sm text-red-500">{formError.email}</span>

  <FloatingLabel
    type="password"
    name="password"
    variant="outlined"
    label="Password"
    onChange={handleChange}
  />
  <span className="text-sm text-red-500">{formError.password}</span>

  <Button type="submit" disabled={!allowSubmit}>
    Sign In
  </Button>
</form>
```

```
const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
  e.preventDefault();
  console.log(formData);
};
```



React Hook Form

Performant, flexible and extensible forms with easy-to-use validation.

Install

```
> npm i react-hook-form
```



Install

```
> npm i joi
```



Install

```
> npm i @hookform/resolvers
```



Sign In

Email

admin@gmail.com

Password

...

"password" must be at least nine characters long and contain an uppercase letter, a lowercase letter, a number and one of the following characters !@#\$\$%^&*~

Sign In

React-hook-form & Joi

- לשם מה נשתמש בספרייה react-hook-form?
- מדוע שלא נעבוד עם טפסים באותה הצורה כמו מקודם?
- מה היא וסון?
- איזה ספריות נצטרך להתקין על הפרויקט שלנו?

JOI



JOI Validations

- מה היא "סכמת ולידציה"?
- מה ניתן להגדיר ב"סכמת ולידציה"?
- כיצד נגדיר regex לצורך ולידציה?
- כיצד נציג הודעת שגיאה משלנו?
- איזה עוד דברים נוכל לבדוק באמצעות שימוש בולידציות JOI?

```
import Joi from "joi";

export const LoginSchema = Joi.object({
  email: Joi.string()
    .email({ tlds: { allow: false } })
    .required(),
  password: Joi.string()
    .ruleset.pattern(
      /((?=.*[0-9]{1})(?=.*[A-Z]{1})(?=.*[a-z]{1})(?=.*[!@#$%^&* -]{1}).{8,20})/
    )
    .rule({
      message:
        '"password" must be at least nine characters long and contain an uppercase letter, a lowercase letter, a number and one of the following characters !@#$%^&*-',
    })
    .required(),
});
```

React-hook-form

```
const initialFormData = {
  email: "",
  password: "",
};

const {
  register,
  handleSubmit,
  formState: { errors },
} = useForm({
  defaultValues: initialFormData,
  mode: "onChange",
  resolver: joiResolver(LoginSchema),
});
```

- איזה משתנים "נפרק" כאשר נשתמש ב-hook זה?
- מה עלינו להעביר לו כערך?
- איזה הגדרות נוכל לשים באובייקט הקונפיגורציה?
- מה המשמעות והשימוש של המשתנה register?
- מה המשמעות והשימוש של המשתנה handleSubmit?
- מה המשמעות והשימוש של המשתנה errors?
- כיצד נוכל לבדוק אם השדות ולידיים (תקינים)?
- מה יחזיק המשתנה errors במידה ויש שגיאות ולידציה?

```
const checkErrors = () => {
  return errors["email"] === undefined && errors["password"] === undefined;
};
```

טפסים – תרגול – חלק ב'

```
<form
  className="flex flex-col gap-4 rounded-lg p-4 shadow-lg"
  onSubmit={handleSubmit(submit)}
>
  <h1 className="text-2xl font-bold text-gray-800">Sign In</h1>
  <FloatingLabel
    type="email"
    variant="outlined"
    label="Email"
    {...register("email")}
    color={errors["email"] ? "error" : "success"}
  />
  <span className="text-sm text-red-500">{errors["email"]?.message}</span>

  <FloatingLabel
    type="password"
    variant="outlined"
    label="Password"
    {...register("password")}
    color={errors["password"] ? "error" : "success"}
  />
  <span className="text-sm text-red-500">
    {errors["password"]?.message}
  </span>

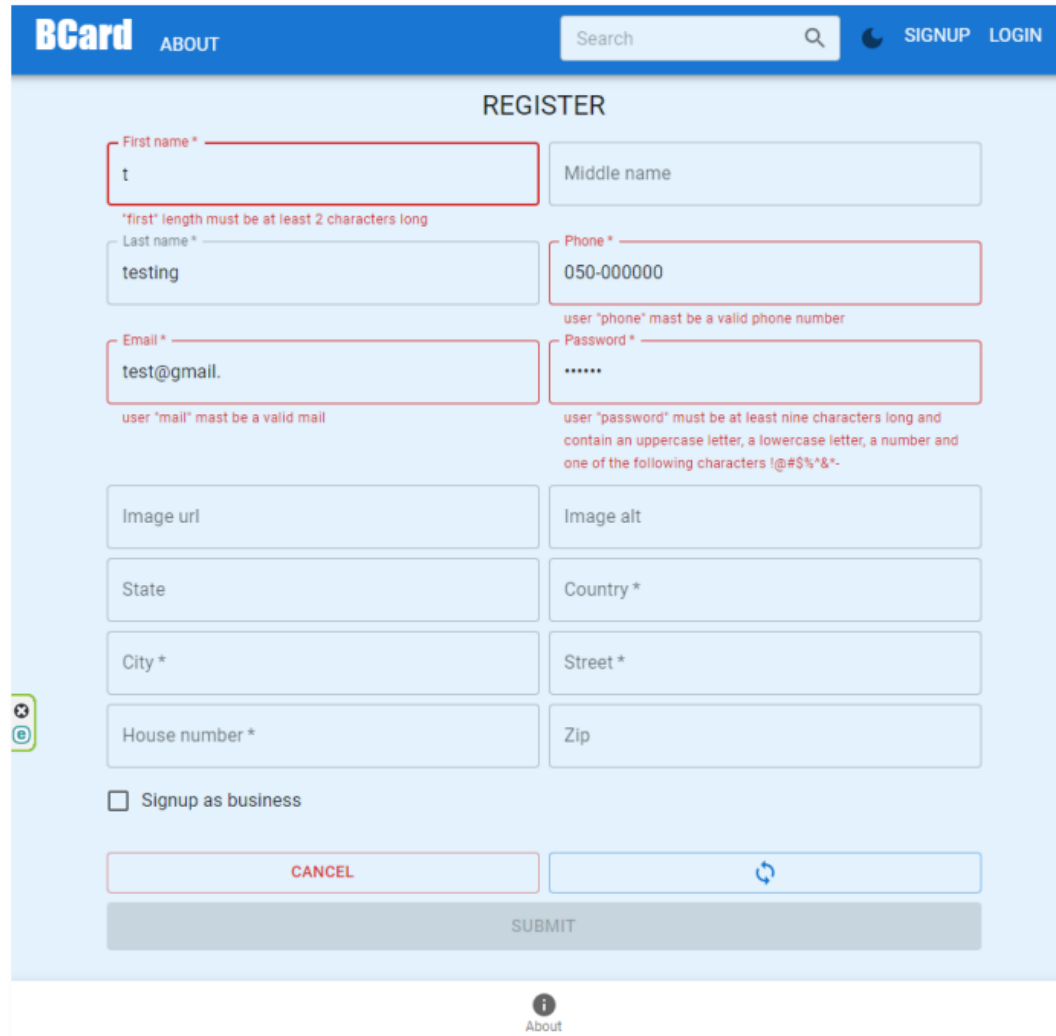
  <Button type="submit" disabled={!checkErrors()}>
    Sign In
  </Button>
</form>
```

- צרו קומפוננטה חדשה בשם SignIn בהתאם לנספח.
- הקומפוננטה תציג טופס התחברות הכולל שדות אימייל ושדה סיסמא תוך שימוש ב-react-hook-form-1 וJoi.
- טפלו בטופס כך שהשדות יעברו ולידציה בכל הקלדת משתמש.
- במידה ויש שגיאת ולידציה, הודעת השגיאה תופיע מתחת לשדה המבוקש.
- אפשרו גישה לכפתור השליחה רק במידה והטופס ולידי.
- במידה והטופס ולידי (תיקני), הדפיסו את תוכנו לקונסול בעת לחיצה על הכפתור.

```
const submit = (form: {email: string, password:string}) => {
  console.log(form);
};
```


שיעורי בית

- פיתחו את פרויקט סוף המודול שלכם.
- צרו טפסי SignUp ו-Register בהתאם למה שלמדנו היום.
- הוסיפו עמודים וקישורים לעמודים אלו מבר הניווט בהתאם.
- בעת לחיצה על כפתור השליחה של הטפסים, בצעו את הפעולה המבוקשת מול ה-API.



BCard ABOUT Search SIGNUP LOGIN

REGISTER

First name * t
'first' length must be at least 2 characters long

Middle name

Last name * testing
user 'mail' must be a valid mail

Phone * 050-000000
user 'phone' must be a valid phone number

Email * test@gmail.
user 'mail' must be a valid mail

Password *
user 'password' must be at least nine characters long and contain an uppercase letter, a lowercase letter, a number and one of the following characters !@#\$%^&*-

Image url Image alt

State Country *

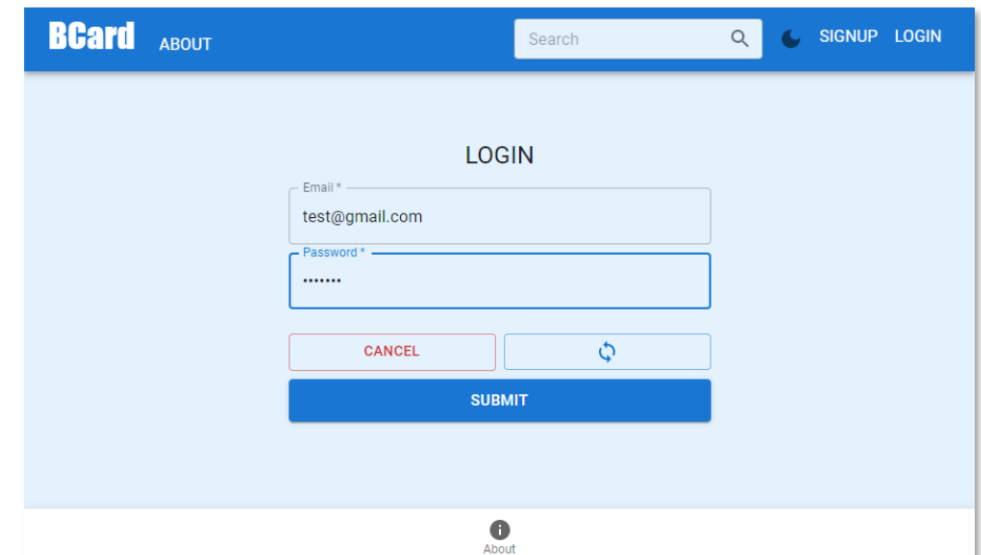
City * Street *

House number * Zip

☐ Signup as business

CANCEL SUBMIT

About



BCard ABOUT Search SIGNUP LOGIN

LOGIN

Email * test@gmail.com

Password *
user 'password' must be at least nine characters long and contain an uppercase letter, a lowercase letter, a number and one of the following characters !@#\$%^&*-

CANCEL SUBMIT

About

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם