Presented by **Michael Romashov**,
Joseph Platt,
Tyler Ortiz,
and **James Zou**

March 5, 2024

TutorTime

The Tutor Scheduling App

01

The Idea

Tutoring Scheduler

Create a Business-to-Customer(B2C) tutoring appointment platform

Students seeking academic help can connect with qualified tutors through an online platform.

Students can schedule one-on-one sessions

Platform streamlines the process of finding, booking and paying for tutoring services.

Custom Services

Allows businesses to customize their:

- Availability
- Service type

Includes account creation

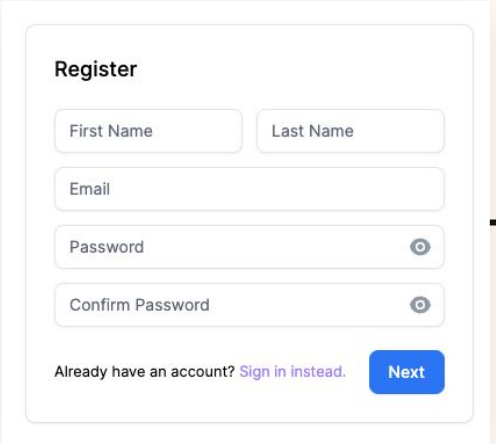
Allow companies to create new services with custom processes

02

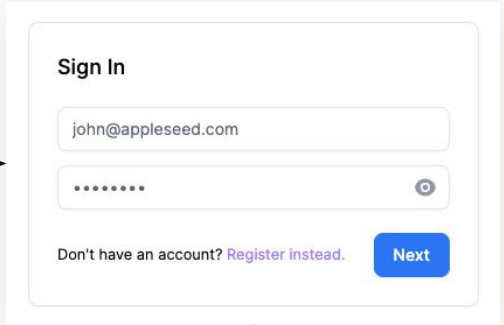
User Flows

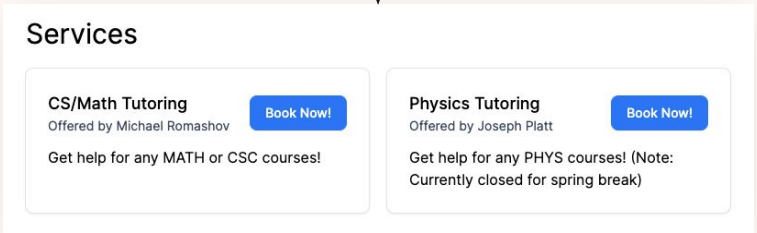
New User Experience

① Account Creation



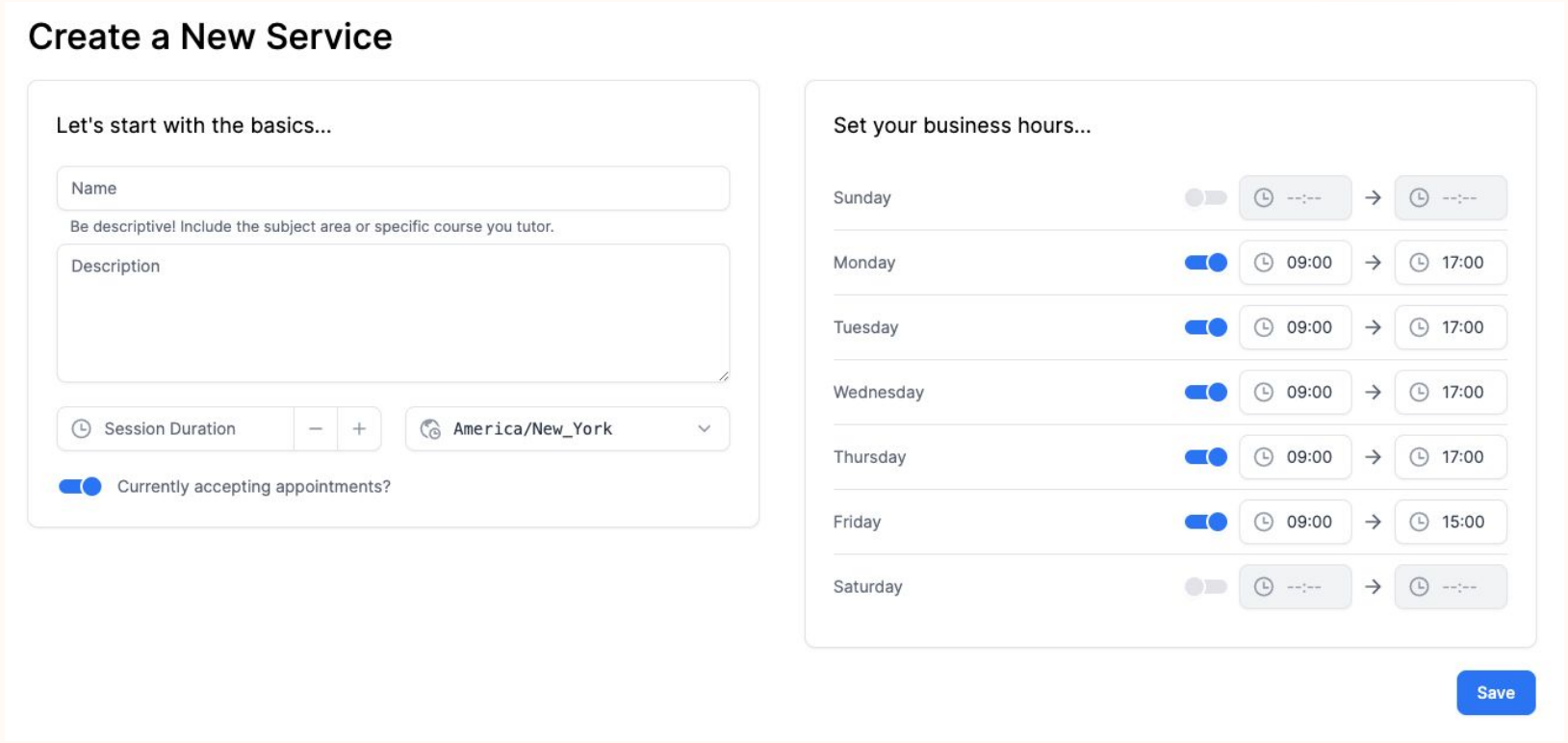
② Authentication





③ Service Browsing

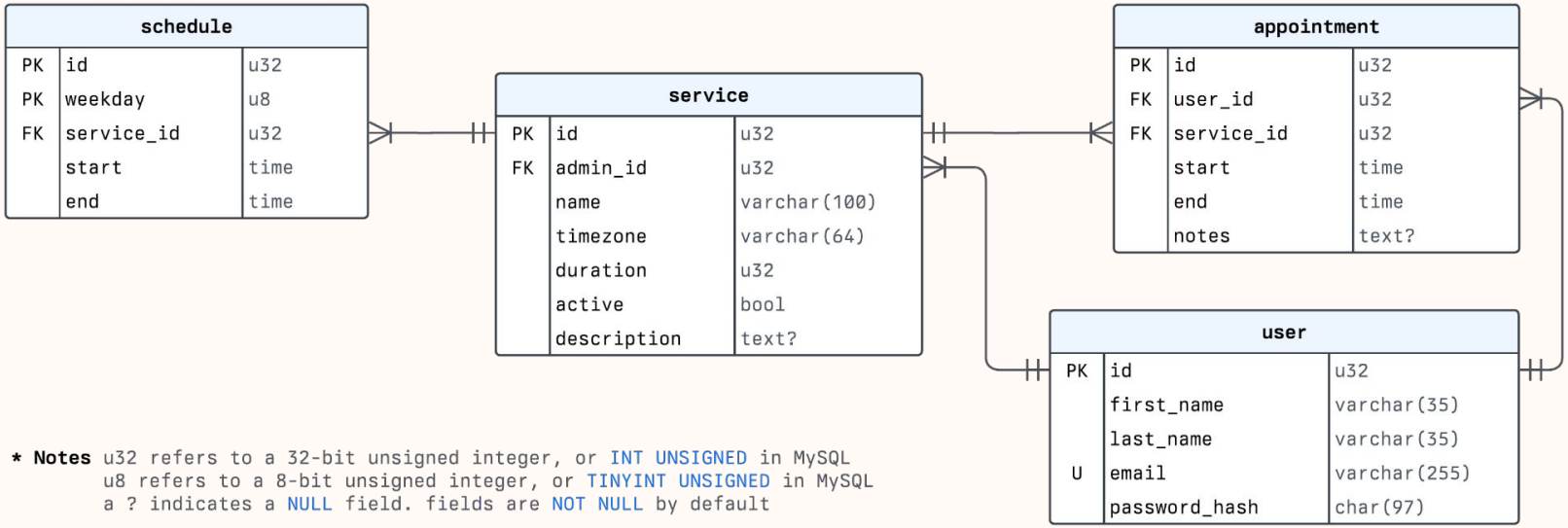
Service Creation



03

Data Modeling

Entity-Relationship Diagram



*** Notes** u32 refers to a 32-bit unsigned integer, or `INT UNSIGNED` in MySQL
u8 refers to a 8-bit unsigned integer, or `TINYINT UNSIGNED` in MySQL
a ? indicates a `NULL` field. fields are `NOT NULL` by default

04

Tech Stack

Database System — **MySQL**

Picking out a database system with the dozens of options we have today is a balancing act between performance, ease-of-use, and scalability.

MySQL performs better than many other systems for smaller databases, while still providing a rich feature set—perfect for us.

Our entire team is also more or less familiar with the engine, which makes development a lot easier.

01-create-tables.sql

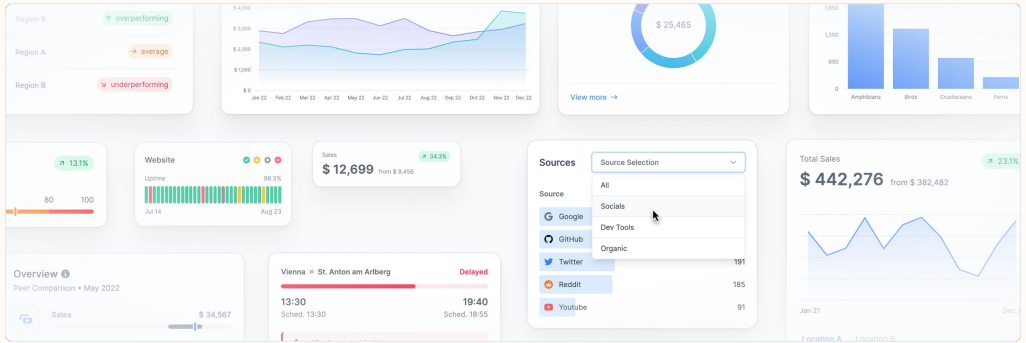
```sql
CREATE TABLE service (
    id          INT UNSIGNED NOT NULL AUTO_INCREMENT,
    name        VARCHAR(100) NOT NULL,
    admin_id    INT UNSIGNED NOT NULL,
    timezone    VARCHAR(64)  NOT NULL,
    duration    INT UNSIGNED NOT NULL,
    active      BOOLEAN      NOT NULL DEFAULT TRUE,
    description TEXT         NULL,

    PRIMARY KEY (id),
    FOREIGN KEY (admin_id) REFERENCES user(id)
);
```

Full-Stack App — **NextJS**

NextJS is a full-stack React framework that allows us to manage frontend and backend code.

We can effortlessly create server-side functions right in our front-end code, while maintaining type safety through **TypeScript**.

We are using a UI component library called **Tremor** to build the interfaces without needing to write our own components.



Development Environment

Docker Compose helps us build and configure our stack with ease and allows us to run all parts of our apps with one single command.

If we ever require a Redis caching layer, or an nginx reverse proxy, or if we need to pick a new database engine entirely, we need only add one section to a configuration file.

docker-compose.yaml

```yaml
services:
  db:
    image: mysql:8.0
    init: true
    restart: always
    ports:
      - 3306:3306
    volumes:
      - ./db-procs:/docker-entrypoint-initdb.d
    environment:
      MYSQL_DATABASE: ${DB_NAME}
      MYSQL_ROOT_PASSWORD: root
      MYSQL_USER: ${DB_USER}
      MYSQL_PASSWORD: ${DB_PASSWORD}
    healthcheck:
      test: mysqladmin ping -h 127.0.0.1 -u
$$MYSQL_USER --password=$$MYSQL_PASSWORD
      interval: 5s
      timeout: 5s
      retries: 20

  web:
    build:
      context: .
      target: dev
    init: true
    restart: always
    depends_on:
      db:
        condition: service_healthy
    ports:
      - 3000:3000
    volumes:
      - .:/app
      - /app/node_modules
      - /app/.next
```

Thank you!

Questions?